
BACHELORARBEIT

Herr
Christian Thormann

**Automatische Optimierung
von Reglerparametern für
einen Positionierantrieb**

Mittweida, 2013

BACHELORARBEIT

Automatische Optimierung von Reglerparametern für einen Positionierantrieb

Autor:

Herr

Christian Thormann

Studiengang:

Elektrotechnik - Automatisierungstechnik

Seminargruppe:

ET09wA-B

Erstprüfer:

Prof. Dr.-Ing. Thomas Beierlein

Zweitprüfer:

Dipl.-Ing. (FH) Jan Färber

Einreichung:

Mittweida, 31.01.2013

Verteidigung/Bewertung:

Mittweida, 2013

BACHELOR THESIS

Automatic optimization of control parameters for a positioning drive

author:

Mr.

Christian Thormann

course of studies:

**Electrical Engineering – Automation
Technology**

seminar group:

ET09wA-B

first examiner:

Prof. Dr.-Ing. Thomas Beierlein

second examiner:

Dipl.-Ing. (FH) Jan Färber

submission:

Mittweida, 31.01.2013

defence/ evaluation:

Mittweida, 2013

Bibliografische Beschreibung:

Thormann, Christian:

Automatische Optimierung von Reglerparametern für einen Positionierantrieb. -
2013. – 41 Seiten, 43 Abbildungen, 5 Tabellen, 2 Anlagen

Mittweida, Hochschule Mittweida, Fakultät Elektro- & Informationstechnik,
Bachelorarbeit, 2013

Referat:

Für das Projekt „Pfeifen-Orgel mit dynamischer Umstimmung“ werden optimale Reglerparameter zur Positionierung der Umstimmelemente benötigt.

Die vorliegende Arbeit befasst sich mit der Entwicklung und Umsetzung eines Algorithmus zur automatischen Optimierung der Reglerparameter. Die Simulation sowie die Einbindung in das bestehende System werden behandelt.

Inhalt

Inhalt	I
Abbildungsverzeichnis	III
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	VII
1 Einleitung.....	1
1.1 Vorwort.....	1
1.2 Aufgabenstellung.....	2
2 Stand der Technik	3
2.1 Client-Aktor-Steuerung.....	3
2.2 Umstimmeinrichtung.....	3
2.3 Mathematische Grundlagen	5
2.3.1 Der Regelkreis	5
2.3.2 Der Regler.....	5
2.3.2.1 P-Glied	5
2.3.2.2 I-Glied	6
2.3.2.3 D-Glied.....	6
2.3.2.4 PID-Regler	7
2.3.3 Qualitätsmerkmal einer Regelstrecke	8
2.4 Optimierungsverfahren	9
2.4.1 Gradientenverfahren	9
2.4.2 Downhill-Simplex-Verfahren	10
2.4.3 Suchverfahren.....	11
3 Simulation.....	13
3.1 Der Regler.....	13
3.2 Die Regelstrecke.....	14
3.3 Das Qualitätsmerkmal.....	14
3.4 Die Optimierungsverfahren.....	15
3.4.1 Das Gradientenverfahren	15
3.4.2 Das Suchverfahren.....	15

3.5	<i>Tests und Ergebnisse</i>	16
4	Implementierung und Test	21
4.1	<i>Umsetzung des Optimierungsalgorithmus in C</i>	21
4.2	<i>Einbindung in Firmware</i>	22
4.2.1	Firmwareversion 1.4.....	22
4.2.2	Firmwareversion 1.50.....	23
4.3	<i>Tests und Ergebnisse</i>	23
4.3.1	Labortest.....	23
4.3.2	Gegenüberstellung der Verfahren	31
4.3.2.1	Umstimmeinrichtung 1	32
4.3.2.2	Umstimmeinrichtung 2	34
4.3.2.3	Umstimmeinrichtung 3	35
4.3.2.4	Umstimmeinrichtung 4	37
4.3.3	Feldtest.....	39
5	Fazit	41
5.1	<i>Ergebnisse</i>	41
5.2	<i>Ausblick</i>	41
Literatur		43
Anlagenverzeichnis		45
	A: Dokumentation zur Simulation Regleroptimierung	A-i
	B: Dokumentation zur CAS-Firmware-Erweiterung 1.4.....	A-v
Selbstständigkeitserklärung		ix

Abbildungsverzeichnis

Abbildung 1: Projektübersicht	1
Abbildung 2: Umstimmeinrichtung	4
Abbildung 3: Regelkreis	5
Abbildung 4: Simulation Regler	13
Abbildung 5: Simulation Regelstrecke	14
Abbildung 6: Simulation Qualitätsmerkmal	14
Abbildung 7: Verhalten der Transferfunktion	16
Abbildung 8: Verhalten eines realen Antriebs	17
Abbildung 9: Verhalten des Sub-Systems Regelstrecke	17
Abbildung 10: Simulationsergebnis der Gradientenanalyse	18
Abbildung 11: Simulationsergebnis der Optimierung mit Suchverfahren.....	19
Abbildung 12: Simulationsergebnis mit Störgröße	20
Abbildung 13: Reglerantwort nach Optimierung mit einem Sprung.....	24
Abbildung 14: Sollwertvorgaben nach Optimierung mit einem Sprung	24
Abbildung 15: Sollwertvorgaben nach Optimierung mit 7 Sprüngen	25
Abbildung 16: 8 Sollwertvorgaben nach Optimierung mit 16 Sprüngen	25
Abbildung 17: 16 Sollwertvorgaben nach Optimierung mit 16 Sprüngen	26
Abbildung 18: 8 Sollwertvorgaben nach Optimierung mit neuem Qualitätsmerkmal	27
Abbildung 19: 16 Sollwertvorgaben nach Optimierung mit neuem Qualitätsmerkmal	27
Abbildung 20: 8 Sollwertvorgaben nach Optimierung mit geteiltem P-Anteil.....	28

Abbildung 21: 16 Sollwertvorgaben nach Optimierung mit geteiltem P-Anteil.....	28
Abbildung 22: 8 Sollwertvorgaben nach Optimierung	29
Abbildung 23: 16 Sollwertvorgaben nach Optimierung	29
Abbildung 24: 8 Sollwertvorgaben nach Optimierung	30
Abbildung 25: 16 Sollwertvorgaben nach Optimierung	30
Abbildung 26: Sollwertvorgaben nach Optimierung mit Nullpunktstart.....	31
Abbildung 27: Sollwertvorgaben nach ARE und Optimierung	31
Abbildung 28: Umstimmeinrichtung 1 Automatische Reglerparameterermittlung.....	32
Abbildung 29: Umstimmeinrichtung 1 Optimierung nach ARE	33
Abbildung 30: Umstimmeinrichtung 1 Nullpunktstart-Optimierung	33
Abbildung 31: Umstimmeinrichtung 2 Automatische Reglerparameterermittlung.....	34
Abbildung 32: Umstimmeinrichtung 2 Optimierung nach ARE	34
Abbildung 33: Umstimmeinrichtung 2 Nullpunktstart-Optimierung	35
Abbildung 34: Umstimmeinrichtung 3 Automatische Reglerparameterermittlung.....	36
Abbildung 35: Umstimmeinrichtung 3 Optimierung nach ARE	36
Abbildung 36: Umstimmeinrichtung 3 Nullpunktstart-Optimierung	36
Abbildung 37: Umstimmeinrichtung 4 Automatische Reglerparameterermittlung.....	37
Abbildung 38: Umstimmeinrichtung 4 Optimierung nach ARE	38
Abbildung 39: Umstimmeinrichtung 4 Nullpunktstart-Optimierung	38
Abbildung 40: Sollwertsprungtest Umstimmelement 1 Automatische Reglerparameterermittlung	39
Abbildung 41: Sollwertsprungtest Umstimmelement 1 Optimierung	39

Abbildung 42: Sollwertsprungtest Umstimmelement 2 Automatische Reglerparameterermittlung.....	40
Abbildung 43: Sollwertsprungtest Umstimmelement 2 Optimierung.....	40

Tabellenverzeichnis

Tabelle 1: Parametrierungszeiten	32
Tabelle 2: Umstimmeinrichtung 1 Reglerparameter	33
Tabelle 3: Umstimmeinrichtung 2 Reglerparameter	35
Tabelle 4: Umstimmeinrichtung 3 Reglerparameter	37
Tabelle 5: Umstimmeinrichtung 4 Reglerparameter	38

Abkürzungsverzeichnis

ARE	Automatische Reglerparameterermittlung
CAN	Controller-Area-Network
CAS	Client-Aktor-Steuerung
MIDI	Musical Instrument Digital Interface
PID-Regler	Proportional-Integral-Differential-Regler
ZAS	Zentrale-Aktor-Steuerung

1 Einleitung

1.1 Vorwort

Die vorliegende Arbeit entstand im Rahmen eines Projektes, in welchem die Pfeifen einer Orgel während des Spiels dynamisch umgestimmt werden können. Diese Umstimmsteuerung ermöglicht es dem Orgelspieler während des Spiels die Stimmung der Orgel zu variieren. Das Projekt ist ein Kooperationsprojekt zwischen der Hochschule Mittweida, dem deutschen Orgelbau A.Vogt GmbH und dpm Identsysteme GmbH. Die Projektübersicht ist in Abbildung 1 zu sehen:

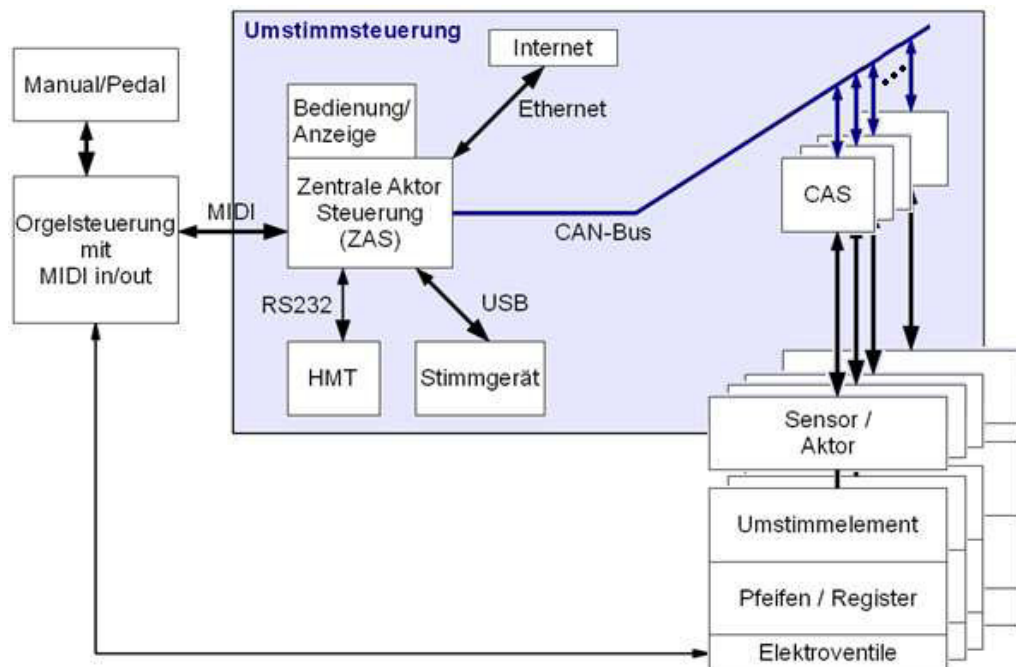


Abbildung 1: Projektübersicht

Hierzu wurde eine Zentrale Akteur Steuerung (ZAS) entwickelt, welche mittels MIDI (Musical Instrument Digital Interface) mit der Orgelsteuerung kommuniziert. Die ZAS steuert via CAN (Controller Area Network) die Client-Aktor-Steuerungen (CAS). Eine CAS ist dafür zuständig vier unabhängige Umstimmeinrichtungen zu steuern. Im Gesamtsystem existieren 78 CAS um bis zu 312 Umstimmeinrichtungen zu bedienen. Eine Umstimmeinrichtung kann ein Umstimmelement an einer Pfeife verstellen, um damit den Ton zu beeinflussen. Dazu verfügt jede Umstimmeinrichtung über einen Lagesensor und einen Linearantrieb. Die Orgel besteht aus 310 Pfeifen mit Umstimmelement. Die Hauptaufgabe der CAS besteht darin, die Umstimmeinrichtungen auf einen von der ZAS vorgegebenen Wert einzuregulieren. Eine Firmware für die CAS, die diese Aufgabe erfüllt, wurde bereits implementiert und erfolgreich getestet [1].

1.2 Aufgabenstellung

Das Umstimmelement wird durch eine Lageregelung positioniert. Diese Regelung ist als PID-Regler bereits in der CAS implementiert. Für die richtige Positionierung sind die Regelparameter entscheidend. Schlecht eingestellte Parameter führen zum Schwingen des Systems, wobei die Umstimmeinrichtungen schlimmstenfalls beschädigt werden können. Ebenso können schlechte Parameter zu bleibenden Abweichungen zwischen dem Soll- und Istwert führen, sodass die Pfeife den falschen Ton spielt. Die Aufgabe besteht darin, ein System zu entwickeln, welches die Regelparameter optimal ermittelt und einstellt. Die Umstimmeinrichtung darf am Ende bei keinem Sollwert schwingen und muss eine minimale Abweichung zwischen Soll- und Istwert aufweisen. Zudem muss sich der Istwert nach spätestens 30 Millisekunden auf den Sollwert eingestellt haben.

In der vorliegenden Arbeit werden Möglichkeiten zur Lösung des Problems vorgestellt. Dazu werden in Kapitel 2 zunächst die vorhandene Technik und die theoretischen Grundlagen aufgezeigt. In Kapitel 3 wird die Simulation des Problems erläutert. Die Implementierung in die existierende Firmware sowie die Tests werden in Kapitel 4 vorgestellt. Abschließend werden die Ergebnisse in Kapitel 5 dargelegt.

2 Stand der Technik

2.1 Client-Aktor-Steuerung

Eine Client-Aktor-Steuerung (CAS) ist eine Platine, welche mittels CAN mit der ZAS kommuniziert. Eine CAS ist dafür zuständig, vier unabhängige Umstimmeinrichtungen zu bedienen. Für jede Umstimmeinrichtung muss eine Lageregelung durchgeführt werden. Diese beruht auf einer Spannungsregelung, das bedeutet die CAS stellt die Aktorspannung so ein, dass der Sensorwert einem vorgegebenen Sollwert entspricht. Zum Einlesen der Sensorspannungen befinden sich vier Operationsverstärkerschaltungen sowie ein vierkanaliger Analog-Digital-Wandler auf der CAS. Die Ausgabe der Stellspannungen erfolgt über einen vierkanaligen Digital-Analog-Wandler. Der Regelalgorithmus ist in einem Microcontroller implementiert.

In der Firmware ist bereits eine Funktion zur Automatischen Reglerparameterermittlung (ARE) enthalten [1: S.23]. Diese Funktion zeichnet die Sprungantwort der Regelstrecke auf und berechnet anhand der Berechnungsvorschrift nach Chien/Hrones/Reswick die Regelparameter [1]. Die Ergebnisse dieser ARE sind jedoch sehr unterschiedlich. Häufig ist eine bleibende Regelabweichung oder Schwingverhalten zu beobachten.

2.2 Umstimmeinrichtung

Die Umstimmeinrichtung besteht aus einem Antriebselement, einem Sensor sowie einem Umstimmelement. Das Umstimmelement wird mittels Tauchspulenantrieb über eine Mechanik bewegt. Dadurch wird der Ton einer Orgelpfeife beeinflusst. Die aktuelle Position wird durch einen Hallsensor erfasst. Der Aufbau ist in Abbildung 2 zu sehen:

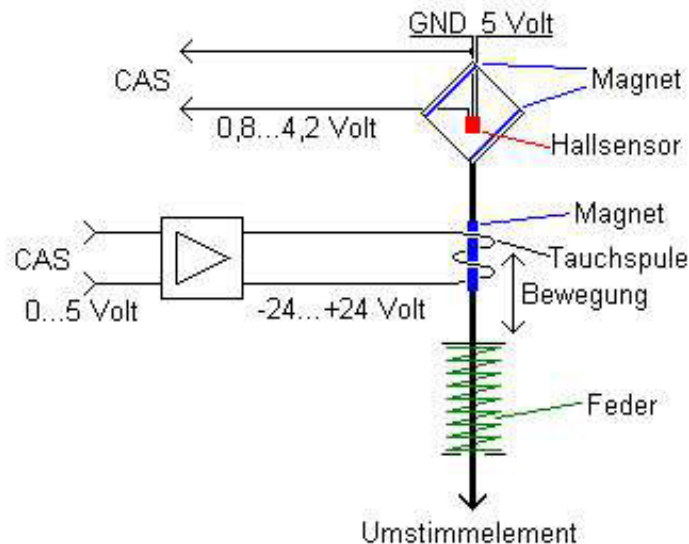


Abbildung 2: Umstimmeinrichtung

Der Hallsensor befindet sich zwischen zwei Magneten, die mit dem Umstimmelement verbunden sind. Die Sensorspannung liegt immer zwischen 0,8 und 4,2 Volt. Die CAS gibt eine Steuerspannung im Bereich 0 bis 5 Volt aus. Die Steuerspannung wird linear auf -24 bis +24 Volt verstärkt. Eine Feder soll die abgeschaltete Umstimmeinrichtung in Mittelposition halten. Durch die nichtlineare Kennlinie des Sensors wird die gesamte Regelstrecke nichtlinear. Zudem beeinflussen Reibung und Massenträgheit das Verhalten der Umstimmeinrichtung.

Zum Testen befanden sich vier verschiedene Umstimmeinrichtungen im Labor.

- Umstimmeinrichtung 1: Diese besitzt zwei Tauchspulen, welche in Reihe geschaltet sind. Bei dieser Umstimmeinrichtung wird an der Orgelpfeife durch Klappen der Ton beeinflusst. Für die Bewegung der Klappen wird viel Kraft benötigt, da die Reibung sehr groß ist.
- Umstimmeinrichtung 2: Diese besitzt ebenfalls zwei Spulen, die jedoch dünner gewickelt sind als bei Umstimmeinrichtung 1. Auch hier beeinflussen Klappen den Ton der Pfeife.
- Umstimmeinrichtung 3: Hierbei handelt es sich um eine Umstimmeinrichtung, die leichter zu bewegen ist als die ersten beiden Umstimmeinrichtungen. Die zu bewegende Masse und die Reibung sind sehr niedrig. Allerdings befindet sich in dem Umstimmelement eine zweite Feder, die das Verhalten beeinflusst. Eine biegsame Lippe verändert hier den Ton.
- Umstimmeinrichtung 4: Auch diese ist leicht zu bewegen, da nur sehr wenig Masse bewegt wird. Durch einen Kegel, welcher in die Pfeife eintaucht, wird der Ton verändert. Da nur dieser Kegel bewegt werden muss, welcher nicht mit der Pfeife verbunden ist, ist die zu bewegende Masse sowie die Reibung niedrig.

2.3 Mathematische Grundlagen

2.3.1 Der Regelkreis

Ein Regelkreis ist ein System, welches eine Regelgröße auf eine Führungsgröße einstellt. Eine Störgröße kann ebenfalls ausgeregelt werden. Abbildung 3 zeigt einen Regelkreis:

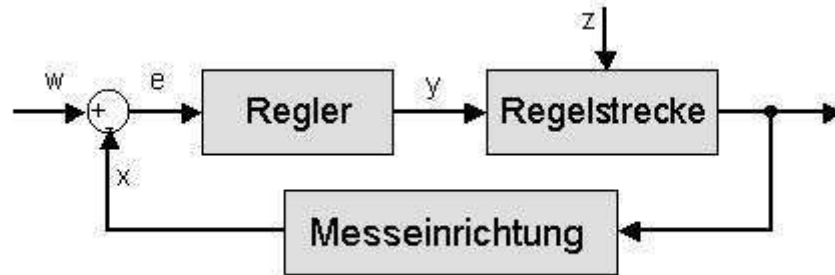


Abbildung 3: Regelkreis

Durch die Messeinrichtung wird permanent die Regelgröße (x) gemessen. Diese wird mit der Führungsgröße (w) verglichen. Das Ergebnis des Vergleichs, die Regelabweichung (e), wird dem Regler zugeführt. Der Regler verarbeitet diese nun und muss durch Verändern der Stellgröße (y) die Regelabweichung verringern oder beseitigen. Der Regler muss in der Lage sein verschiedene Störgrößen (z) auszuregeln.

2.3.2 Der Regler

Der PID-Regler setzt sich aus einem P-, I- und D-Glied zusammen [2]. Betrachtet wurde nur der PID-Regler, da zum Regeln der Umstimmereinrichtung ein stetiger Regler benötigt wird. Dieser Regler lässt sich zudem einfach umsetzen.

2.3.2.1 P-Glied

Das proportionale Übertragungsglied verstärkt die Eingangsgröße um einen konstanten Faktor. Das Verhältnis zwischen Ein- und Ausgangsgröße errechnet sich folgendermaßen [2: S.89]:

$$x_a(t) = K_P x_e(t) \quad (2 - 1)$$

Als Übertragungsfunktion ergibt sich:

$$G(s) = K_P \quad (2 - 2)$$

Das P-Glied kann als reiner Regler eingesetzt werden. Anwendung findet dieser bei Regelstrecken, bei denen eine bleibende Regelabweichung nicht stört. Der reine P-Regler ist sehr dynamisch, da auf eine Regelabweichung (Eingangsgröße) sehr schnell reagiert wird.

2.3.2.2 I-Glied

Das I-Glied integriert die Regelabweichung und verstärkt die Summe. Im Zeitbereich wird das durch folgende Beziehung beschrieben [2: S.90]:

$$x_a(t) = \frac{1}{T_I} \int_0^t x_e(\tau) d\tau \quad (2 - 3)$$

Die Übertragungsfunktion lautet wie folgt:

$$G(s) = \frac{1}{sT_I} \quad (2 - 4)$$

Das I-Glied kann als reiner Regler genutzt werden. Dieser Regler schafft es, eine bleibende Regelabweichung vollständig auszuregulieren, ist jedoch langsam. Anwendung findet der reine I-Regler bei Regelstrecken, die keine hohe Dynamik aufweisen. Die Regelstrecke darf keine Verzögerung aufweisen, da dies zu instabilem Verhalten führen würde.

2.3.2.3 D-Glied

Das D-Glied differenziert die Eingangsgröße und errechnet so die Ausgangsgröße. Das Verhältnis errechnet sich folgendermaßen [2: S.90]:

$$x_a(t) = T_D \frac{d}{dt} x_e(t) \quad (2 - 5)$$

Es ergibt sich folgende Übertragungsfunktion:

$$G(s) = sT_D \quad (2 - 6)$$

Das D-Glied kann nicht als reiner Regler eingesetzt werden. Es kann nur in Kombination mit P- oder I-Glied genutzt werden. Das D-Glied reagiert sehr schnell auf Veränderungen der Eingangsgröße und ist daher sehr dynamisch.

2.3.2.4 PID-Regler

Der PID-Regler ist die Kombination des P-, I- und D-Glieds. Dadurch werden die Regleigenschaften vereint. Das Verhältnis zwischen Ein- und Ausgangsgröße ergibt sich aus der Summe der einzelnen Übertragungsfunktionen (Formel 2-2, 2-4, 2-6) [2: S.124]:

$$G(s) = K_p + \frac{1}{sT_I} + sT_D \quad (2 - 7)$$

Zur Vereinfachung werden folgende Größen eingeführt:

$$K_p = K_R \quad \text{Verstärkungsfaktor}$$

$$T_I = \frac{K_p}{K_i} \quad \text{Integralzeit}$$

$$T_D = \frac{K_d}{K_p} \quad \text{Differenzialzeit}$$

Durch diese Vereinfachung ergibt sich folgende Übertragungsfunktion:

$$G(s) = K_R \left(1 + \frac{K_i}{s} + K_d s \right) \quad (2 - 8)$$

Der zeitliche Verlauf der Ausgangsgröße ergibt sich nach Formel 2-8 folgendermaßen:

$$x_a(t) = K_R x_e(t) + K_i \int_0^t x_e(\tau) d\tau + K_d \frac{d}{dt} x_e(t) \quad (2 - 9)$$

Der Faktor K_R wird im weiteren als P-Anteil bezeichnet. K_i wird als I-Anteil bezeichnet und K_D als D-Anteil.

Um die Reglergleichung (2-9) als digitalen Regler zu implementieren muss diese Gleichung durch Diskretisierung eine Differenzengleichung transformiert werden. Hierbei wird das Integral durch eine Summe ersetzt und der Differenzialquotient durch eine Differenzbildung der beiden letzten Eingangswerte. T_0 steht hierbei für die Abtastzeit. Es ergibt sich folgende Reglergleichung [3: S.108]:

$$x_a(t_k) = K_R x_e(t_k) + T_0 K_i \sum_{i=0}^{k-1} x_e(t_i) + \frac{1}{T_0} K_d (x_e(t_k) - x_e(t_{k-1})) \quad (2 - 10)$$

In der CAS-Firmware [1] wurde der Regelalgorithmus nach folgendem Prinzip implementiert. Die Umsetzung ist als Pseudocode dargestellt:

$$esum = esum + e$$

$$y = K_p * e + K_i * T_0 * esum + K_d * \frac{e - e_{alt}}{T_0}$$

$$e_{alt} = e$$

Der Aufruf des Regelalgorithmus erfolgt zyklisch jede Millisekunde. Die Abtastzeit T_0 ist daher eine Millisekunde.

2.3.3 Qualitätsmerkmal einer Regelstrecke

Eine Regelstrecke ist ein physikalisches System, welches mittels einer Stellgröße beeinflusst wird. Durch eine Messeinrichtung wird ein Istwert der Regelstrecke erfasst. Für das vorliegende Problem wird ein Merkmal der Regelstrecke benötigt, welches für einen eingestellten PID-Regler eine Aussage zur Qualität trifft. Dieses Merkmal muss dabei die bleibende Regelabweichung sowie schwingendes Verhalten erfassen. Die gemessene Regelabweichung, der Vergleich zwischen Führungsgröße (Sollwert) und Regelgröße (Istwert) kann als Qualitätsmerkmal benutzt werden. Eine bleibende Abweichung würde permanent als Eingangssignal am Regler anliegen, sodass dieser Wert gemessen werden kann. Durch den I-Anteil wird sich der Wert jedoch 0 annähern, daher genügt eine Momentaufnahme nicht zur Qualitätscharakterisierung. Wird die Eingangsgröße (Regelabweichung) allerdings über der Messzeit integriert, lässt sich anhand des Ergebnisses abschätzen, wie gut die Reglerparameter sind. Ist dieser Wert annähernd 0, so existiert eine kleine bis keine Regelabweichung. Falls das Ergebnis konstant ist, so hat sich die Regelgröße auf die Führungsgröße eingestellt. Sollte dieser Wert jedoch konstant steigen, so liegt eine bleibende Regelabweichung vor, die nicht ausgeglichen wird. Daher ergibt sich folgende Formel:

$$Q = \int_0^{\infty} (\text{Sollwert}(\tau) - \text{Istwert}(\tau)) d\tau \quad (2 - 11)$$

Da bei den vorliegenden Regelstrecken der Sollwert kleiner sein kann als der Istwert, könnten in Formel 2-11 ebenso negative Werte entstehen. Bei einer periodisch schwingenden Reglerantwort wäre $Q = 0$, da sich alle positiven und negativen Werte aufheben. Um diesen Effekt zu vermeiden, wird die Regelabweichung quadriert, damit alle Werte positiv sind. Dadurch werden größere Abweichungen auch mehr bestraft als kleine. Dies ist in folgender Gleichung zu sehen:

$$Q = \int_0^{\infty} (\text{Sollwert}(\tau) - \text{Istwert}(\tau))^2 d\tau \quad (2 - 12)$$

Je kleiner die bleibende Regelabweichung, umso kleiner ist der Wert Q . Ebenso führt ein steilerer Anstieg der Reglerantwort zu einem kleineren Q . Schwingt das System jedoch, muss der Qualitätswert deutlich größer sein als bei einem stabilen System. Daher wird die Ableitung der Regelgröße zur Formel 2-12 addiert. Die Ableitung muss quadratisch einbezogen werden, da eine abgeleitete Schwingung wieder eine Schwingung ergibt, welche integriert Null ergeben würde. Formel 2-13 zeigt das Qualitätsmerkmal:

$$Q = \int_0^{\infty} ((\text{Sollwert}(\tau) - \text{Istwert}(\tau))^2) d\tau + \int_0^{\infty} \left(\left(\frac{d}{d\tau} \text{Sollwert}(\tau) - \text{Istwert}(\tau) \right)^2 \right) d\tau \quad (2 - 13)$$

Die Formel 2-13 muss zur Nutzung im Controller diskretisiert werden. Hierzu werden die Integrale in Summen umgewandelt und der Differenzialquotient in eine Differenz. Es kann keine unendliche Erfassungszeit realisiert werden.

$$Q = \sum_{t=0}^n [(e_t)^2] + \sum_{t=0}^n [(e_t - e_{t-1})^2] \quad (2 - 14)$$

$$e_t = \text{Sollwert}(t) - \text{Istwert}(t)$$

Für die Variable n in Formel 2-14 muss die Erfassungszeit eingesetzt werden.

2.4 Optimierungsverfahren

2.4.1 Gradientenverfahren

Das Gradientenverfahren ist ein Optimierungsverfahren, welches für einen Punkt der mehrdimensionalen Funktion den Gradienten (den steilsten Abstieg) bestimmt. Es ist für reellwertige, differenzierbare Funktionen anwendbar. Der Gradient wird durch partielles Ableiten bestimmt. Er errechnet sich wie folgt [4: S.97]:

$$\text{grad}(f) = \nabla f = \frac{\partial f}{\partial x} \hat{e}_x + \frac{\partial f}{\partial y} \hat{e}_y + \frac{\partial f}{\partial z} \hat{e}_z \quad (2 - 15)$$

Der Eintrag $\frac{\partial f}{\partial x}$ steht für die partielle Ableitung der Funktion nach dem Parameter x . \hat{e}_x steht für die Multiplikation mit dem Einheitsvektor in x -Richtung. Das Ergebnis ist ein Vektor, der in Richtung des steilsten Anstiegs zeigt. Da das Qualitätsmerkmal jedoch minimiert werden soll, wird das Ergebnis von dem Startwert abgezogen. Falls der Ergebnisvektor nicht null ist, so kann eine neue Gradientenanalyse durchgeführt werden. Ist der Ergebnisvektor null, so wurde ein Extremwert gefunden. Der errechnete Punkt ist dann ein lokales Minimum. Damit dieses Verfahren funktioniert, muss eine Funktion genutzt werden, die stetig differenzierbar ist. Ebenso darf der Gradient im Startpunkt nicht null sein, da das Verfahren sonst nicht funktioniert.

Für das vorliegende Problem ergibt sich eine Funktion, in der der Funktionswert das Qualitätskriterium ist und die Parameter die Reglerparameter sind. Da die Funktionsgleichung allerdings von der Regelstrecke abhängt und deren Verhalten unbekannt ist, kann nicht davon ausgegangen werden, dass diese Funktion differenzierbar ist. Das Verhalten der Regelstrecke hängt von dem Aufbau der Umstimmeinrichtung ab, sowie von der Sensor-kennlinie und der verwendeten Feder. Daher kann der Gradient nur numerisch bestimmt werden. Dazu muss zunächst ein Startpunkt, bestehend aus den PID-Parametern (P-Anteil, I-Anteil und D-Anteil), gewählt werden, für den der Funktionswert (Qualitätswert) anhand eines Sprunges bestimmt wird. Anschließend wird der erste Parameter um einen Einheitswert erhöht und der Funktionswert bestimmt. Als Einheitswert ist ein fester Wert zu wählen, beispielsweise 1,0. Anhand der Differenz der Funktionswerte wird der Anstieg errechnet. Dies muss für alle drei Parameter wiederholt werden, wobei die anderen beiden Parameter auf den Ursprungswert zurückgesetzt werden müssen. Die drei Anstiegswerte bilden nun einen Vektor, der in Richtung des maximalen Anstiegs zeigt. Dieser muss nun von dem Startpunkt subtrahiert werden, um in Richtung des steilsten Abstiegs zu gehen. Dies wiederholt man nun so oft, bis keine Verbesserung mehr möglich ist.

2.4.2 Downhill-Simplex-Verfahren

Das Downhill-Simplex-Verfahren, auch bekannt als Verfahren von Nelder und Mead [4: S.21], ist ein Optimierungsverfahren, welches ohne Ableitung der Funktion arbeitet. Am Beispiel einer dreidimensionalen Funktion $z = f(x, y)$ soll das Prinzip erläutert werden. Zu Beginn wird ein Startpunkt gewählt, dessen Funktionswert bestimmt wird. Ausgehend vom Startpunkt wird nun der erste Parameter (x) um einen bestimmten Faktor vergrößert und der Funktionswert bestimmt. Dies geschieht ebenso für den zweiten Parameter, wobei der erste Parameter auf den Ursprungswert zurückgesetzt wird. Aus diesen drei Punkten wird nun der beste und der schlechteste bestimmt. Falls der beste Wert nicht ausreichend ist, wird anhand der anderen beiden Punkte eine Geradengleichung errechnet. Der schlechteste Punkt wird nun am Schwerpunkt der Geraden reflektiert. Falls sich der Funktionswert verbessert, wird der Punkt weiter in Reflektionsrichtung verschoben, bis keine Verbesserung mehr zu beobachten ist. Anschließend werden wieder alle Werte verglichen und der schlechteste Wert wird wieder gespiegelt. Das geschieht so lange, bis ein Funktionswert ausreichend ist. Der resultierende Punkt ist ein lokales Minimum. Für dieses Ver-

fahren ist keine Ableitung notwendig, die zu testende Funktion muss nicht stetig differenzierbar sein.

2.4.3 Suchverfahren

Eine andere Möglichkeit zur Optimierung ist ein an das Downhill-Simplex-Verfahren angelehntes Verfahren. Hierbei werden ein Startpunkt und alle umliegenden Punkte betrachtet. Um einen umliegenden Punkt zu betrachten, wird einer der Parameter um eine feste Schrittweite erhöht oder verringert. Durch Vergleichen der Punkte wird der beste ermittelt und als neuer Startpunkt gesetzt. Anschließend werden wieder alle umliegenden Punkte, mit Ausnahme des Startpunktes, da dessen Wert ja bereits bekannt ist, betrachtet. Ebenso wird der Punkt, der im vorherigen Durchlauf Startpunkt war, nicht betrachtet, da dieser schlechter war. Sind alle umliegenden Punkte schlechter als der Startpunkt, so wurde eine optimale Lösung gefunden. In diesem Fall kann das Verfahren mit einer kleineren Schrittweite neu gestartet werden.

Vorteil dieses Verfahrens ist, dass es ableitungsfrei ist. Nachteil ist die längere Konvergenzzeit gegenüber anderen Verfahren, da sehr viele Punkte betrachtet werden müssen, um eine Lösung zu finden. Zudem kann nicht sichergestellt werden, dass die gefundene Lösung die global beste Lösung ist.

3 Simulation

Die Simulation wurde mit MATLAB Simulink Version 7.10.0.499 (R2010a) durchgeführt. In der Simulation wird der Regelkreis, bestehend aus Regler, Regelstrecke sowie Sensorik, nachgebildet. Zudem wurden Verfahren zur Optimierung des Reglers simuliert. In Anlage A befinden sich eine Abbildung der Gesamtsimulation (Seite A-i) sowie das Stateflow zur Optimierung (Seite A-iii) aus der Dokumentation zur Simulation [6].

3.1 Der Regler

Für die Simulation war es nicht möglich, den PID-Controller-Baustein aus der Simulink-Bibliothek zu verwenden, da mit diesem nicht während der Laufzeit die PID-Parameter geändert werden können. Daher wurde ein neuer Regler-Baustein entworfen, welcher das Ändern der Parameter während der Laufzeit zulässt und diese zurückgibt. Abbildung 4 zeigt den Regler:

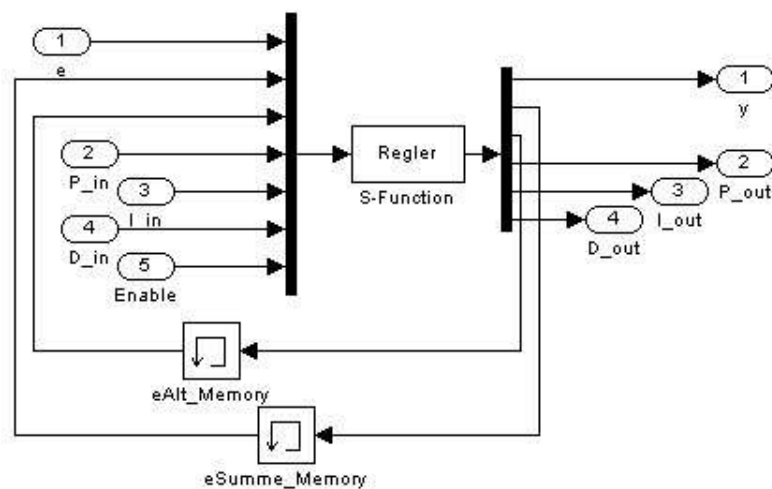


Abbildung 4: Simulation Regler

Die S-Function beinhaltet den Regelalgorithmus. Über die Eingänge P_{in} , I_{in} und D_{in} können die Regelparameter eingegeben werden. P_{out} , I_{out} und D_{out} geben die aktuellen Parameter zurück. Durch die Variable *Enable* wird der Regler durch Anlegen des Wertes 1 aktiviert. Das Anlegen eines Wertes 0 schaltet den Regler ab. Die Regelabweichung muss am Eingang e angelegt werden. Der Ausgang y gibt die Stellgröße zurück.

3.2 Die Regelstrecke

Das Modell der Regelstrecke soll die Umstimmereinrichtung sowie die Sensorik simulieren. Um das Antriebsverhalten zu simulieren, wurde eine Transferfunktion verwendet. In der Lookup-Tabelle wurde die Sensorkennlinie nachgebildet. In Abbildung 5 ist die verwendete Regelstrecke zu sehen:

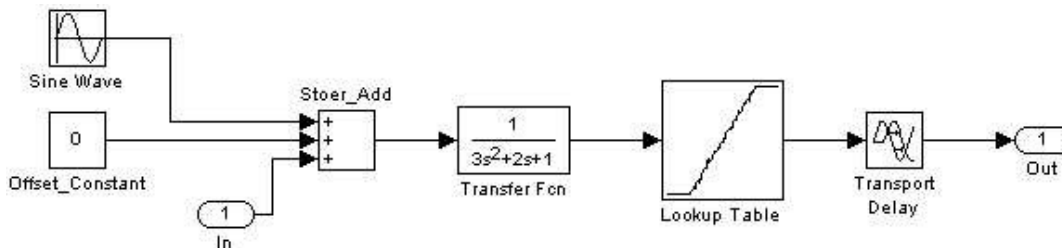


Abbildung 5: Simulation Regelstrecke

Mittels *Sine Wave* und *Offset_Constant* kann eine Störgröße simuliert werden. Der *Transport Delay*-Baustein simuliert die mechanische Trägheit.

3.3 Das Qualitätsmerkmal

Um die Qualität der Parameter zu bestimmen, wurde das SubSystem *FlaecheAufzeichnen* entworfen. Das verwendete SubSystem zur Qualitätserrechnung ist in Abbildung 6 zu sehen:

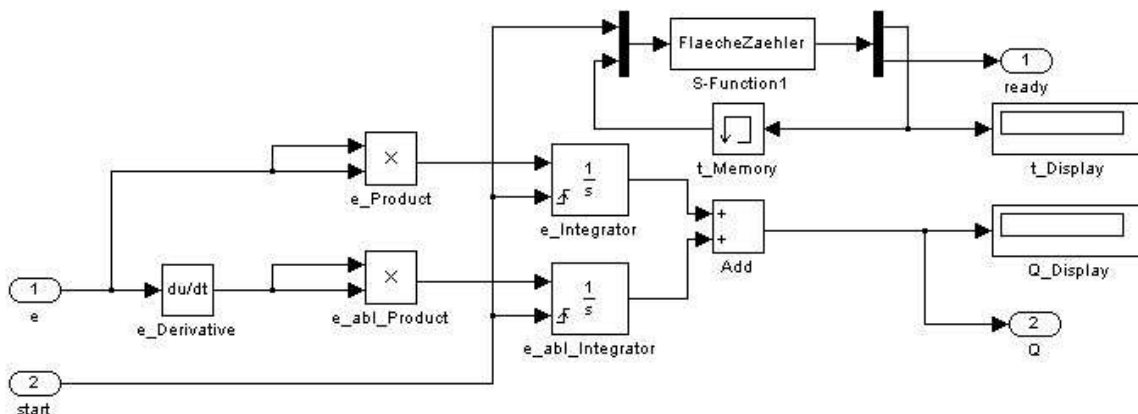


Abbildung 6: Simulation Qualitätsmerkmal

Die Regelabweichung am Eingang e wird mittels $e_Product$ quadriert und durch $e_Integrator$ integriert. Die mittels $e_Derivative$ gebildete Ableitung wird ebenso quadriert und integriert. Dadurch wird die Formel 2-13 nachgebildet. Eine steigende Flanke am Eingang *start* aktiviert die Messung, zudem werden dadurch die Integrationsbausteine genullt, um eine Startbedingung zu schaffen. Die S-Funktion *FlaecheZaeher* zählt nun die Aufrufzyklen bis zum Wert 200. Danach wird der Ausgang *ready* auf den Wert 1 gesetzt.

Dadurch wird eine Messzeit realisiert, da die Funktion nicht unendlich lange messen soll. Am Ausgang *Q* kann nun der Qualitätswert ausgelesen werden.

3.4 Die Optimierungsverfahren

Die Optimierung wurde als Stateflow realisiert, da hiermit eine Ablaufsteuerung umgesetzt werden kann. Ein Stateflow ermöglicht die Simulation von Zustandsautomaten mittels eines Zustandsübergangsdiagramms. In Anlage A ist der Stateflow zur Optimierung (Suchverfahren) abgebildet (Seite A-iii).

3.4.1 Das Gradientenverfahren

Das Gradientenverfahren wurde bereits in Kapitel 2.4.1 beschrieben. Zu Beginn werden als Initialisierung die Standardparameter für den Testsprung und die Differenz festgelegt. Die aktuellen PID-Parameter werden eingelesen. Ebenso wird der *sprungvon*-Wert als Sollwert zurückgegeben. Nach dem Startimpuls wird der aktuelle Qualitätswert gemessen. Dazu wird der *sprungnach*-Wert als Sollwert gesetzt und die Ausgangsvariable *rec* auf 1 gesetzt, um die Messung zu starten. Sobald das *FlächeAufzeichnen*-SubSystem *ready* aktiviert, wird der Qualitätswert eingelesen. Anschließend wird der Sollwert auf *sprungvon* zurückgesetzt. Dieser Ablauf findet nun je einmal für höheren P-Anteil, höheren I-Anteil und für höheren D-Anteil statt. Als Schrittweite wurde 1,0 gewählt. Anschließend wird anhand der vier gemessenen Werte eine numerische Gradientenanalyse durchgeführt, nach der die PID-Parameter neu eingestellt werden.

3.4.2 Das Suchverfahren

In Kapitel 2.4.3 wurde dieses Verfahren erläutert. Bei diesem Verfahren wird ein Startpunkt betrachtet und für jeden Parameter ein Punkt in dem der Parameter um eine feste Schrittweite vergrößert und verkleinert wurde. Als Anfangsschrittweite wurde 1,0 gewählt. Die weiteren Schrittweiten lauten 0,5 , 0,2 sowie 0,1. Für das vorliegende Problem müssen demnach der Startpunkt und sechs weitere Punkte je Durchgang gemessen werden. Aus diesen sieben Punkten wird der kleinste Wert bestimmt und als neuer Startpunkt festgelegt. Nun werden nur noch alle umliegenden Punkte betrachtet, außer dem vorherigen Startpunkt, da dieser schlechter gewesen ist. Dies geschieht so lange, bis alle umliegenden Punkte schlechter sind. In diesem Fall wird die Schrittweite auf den nächsten Wert verkleinert und das Verfahren startet erneut. Falls die Schrittweite minimal war, so ist das Verfahren beendet.

3.5 Tests und Ergebnisse

Um die Verfahren zu testen wurde die Transfer-Funktion

$$X(s) = \frac{1}{3s^2 + 2s + 1} \quad (3 - 1)$$

gewählt. Im Zeitbereich ergibt sich durch eine Laplace-Rücktransformation folgende zeit-kontinuierliche Funktion:

$$x(t) = \frac{\sqrt{2}}{2} * \sin\left(\frac{\sqrt{2} * t}{3}\right) * e^{\frac{-t}{3}} \quad (3 - 2)$$

Gewählt wurde diese Transferfunktion, da diese das Verhalten der Antriebe sehr gut nachgebildet. Abbildung 7 zeigt das Verhalten der Transferfunktion und Abbildung 8 das Verhalten eines realen Antriebs ([1]: S.21) nach einer Stellgrößenänderung:

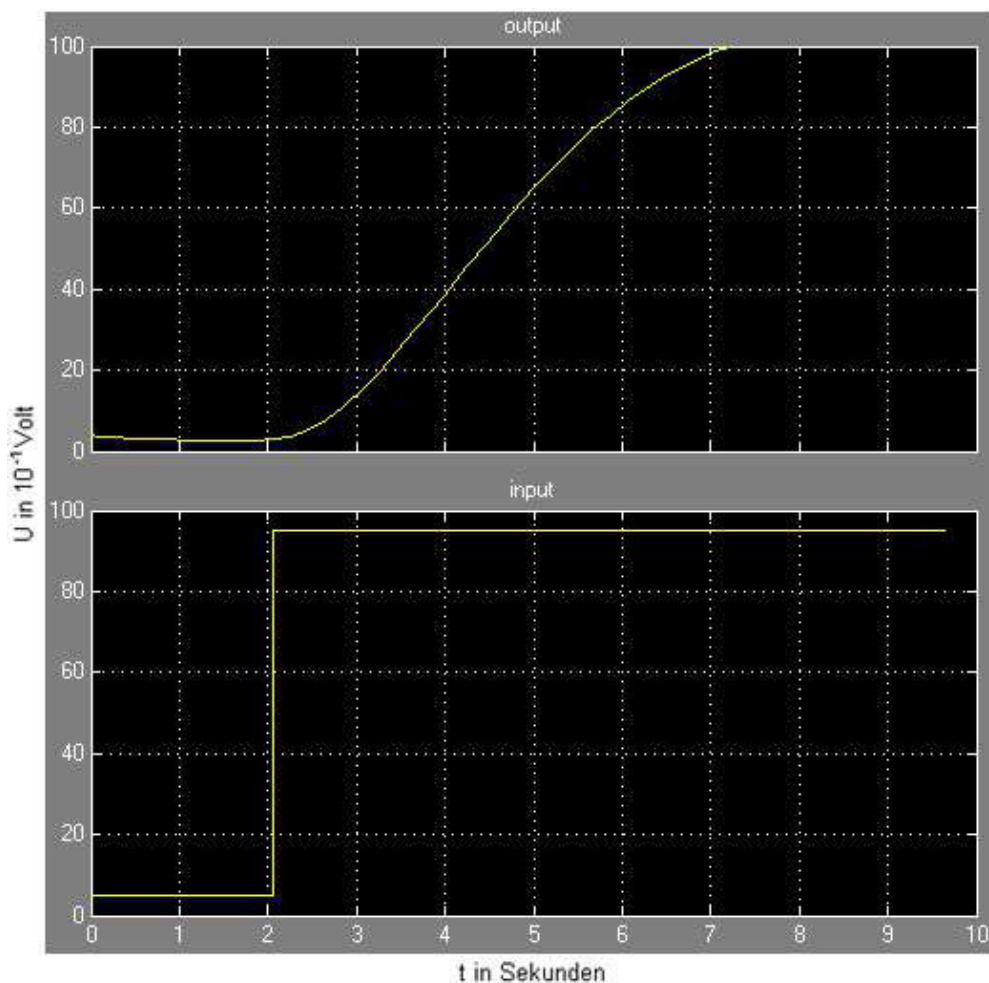


Abbildung 7: Verhalten der Transferfunktion

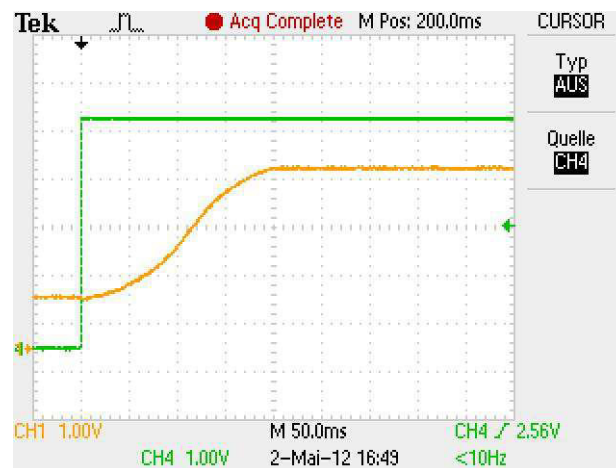


Abbildung 8: Verhalten eines realen Antriebs

In Abbildung 8 wurde mit Kanal 1 (gelb) der Sensorwert und mit Kanal 4 (grün) die Steuerspannung aufgezeichnet. Zu sehen ist, dass in beiden Fällen der Sensorwert (in der Simulation der output) der Vorgabe nur langsam folgt. Die Sensorkennlinie wurde durch eine Lookup-Table nachgebildet. In dieser Tabelle wurden auch die elektrischen Grenzen des Sensors nachgebildet. Diese sind in Abbildung 8 zu sehen, da die Sensorspannung nur im Bereich von 1 Volt bis 3,7 Volt arbeitet. Ein Transport-Delay-Baustein wurde zur Nachbildung der mechanischen Trägheit eingebaut. Als Verzögerungszeit wurde eine Millisekunde gewählt. Abbildung 9 zeigt das endgültige Verhalten des SubSystems Regelstrecke:

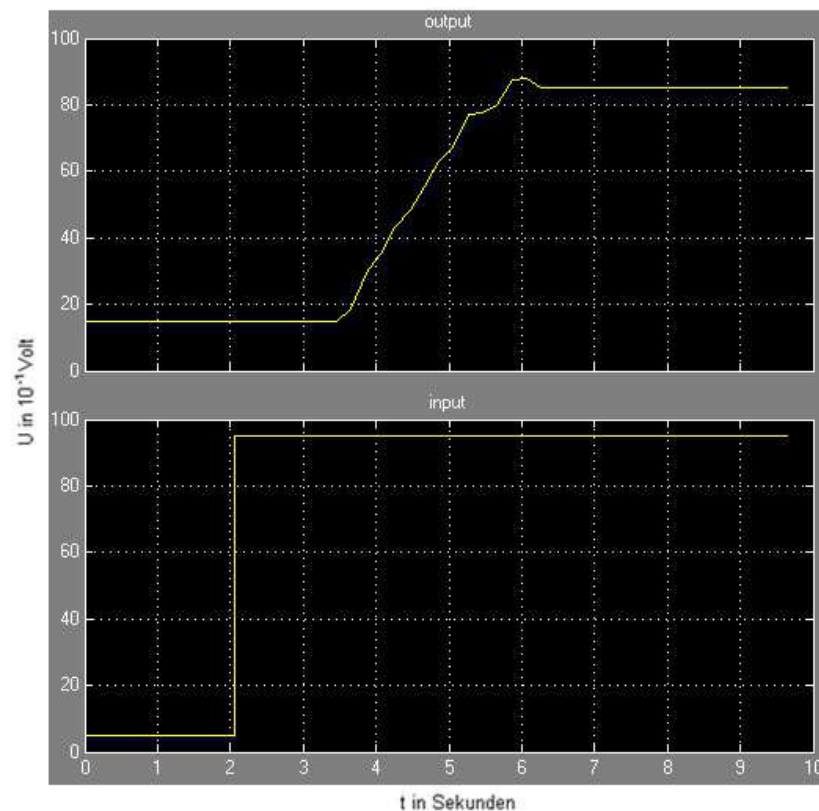


Abbildung 9: Verhalten des Sub-Systems Regelstrecke

Getestet wurde zunächst das Gradientenverfahren (Kapitel 3.4.1). Die Regelparameter waren anfangs alle null. Nach der Berechnung wurde ein Testsollwertsprung durchgeführt, welcher in Abbildung 10 zu sehen ist:

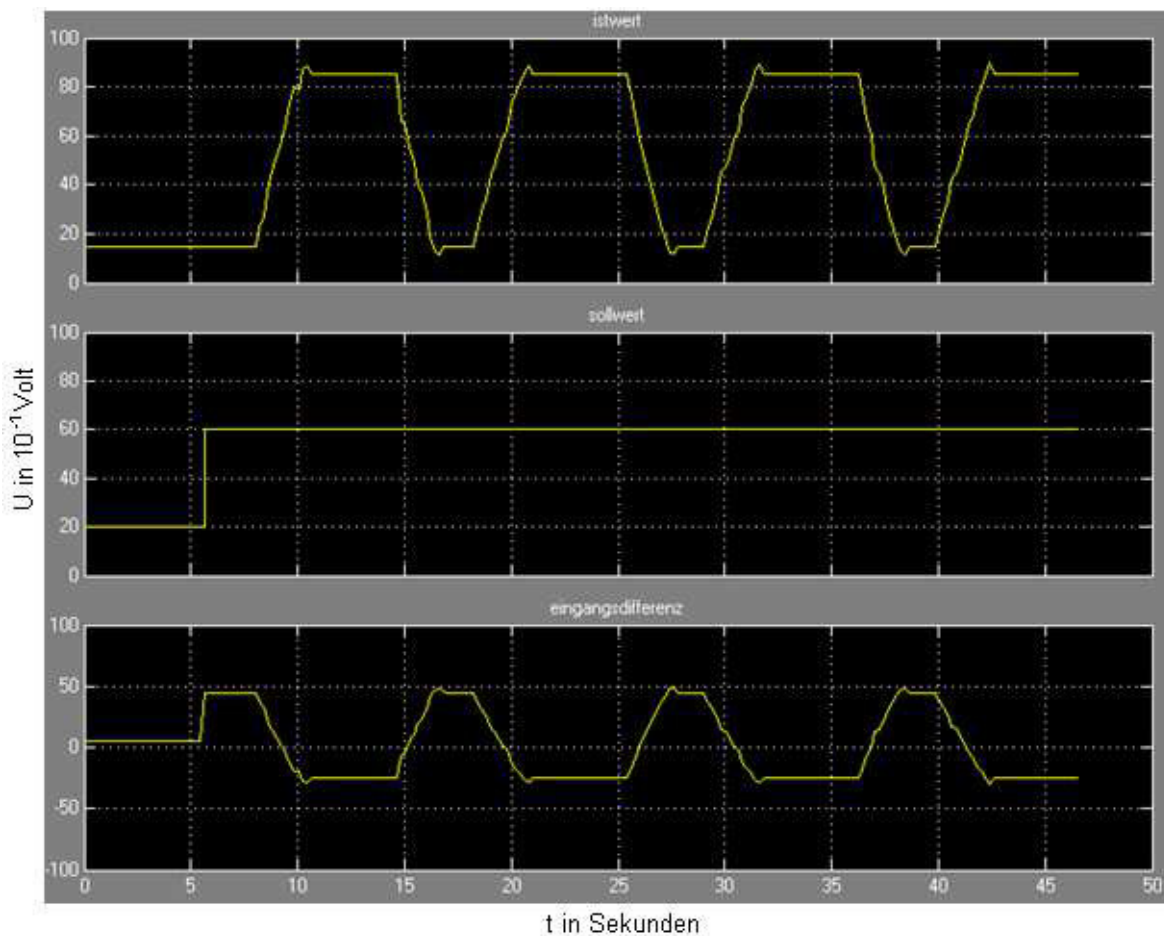


Abbildung 10: Simulationsergebnis der Gradientenanalyse

Das Ergebnis zeigt, dass die Stellgröße schwingt und sich so der Sollwert nicht einstellt. Das Gradientenverfahren errechnet einen Vektor, der in die Richtung des steilsten Abstiegs zeigt. Dieser Vektor wird zu dem Startpunkt addiert, um einen neuen Parametersatz zu bestimmen. Die Tatsache, dass die mathematische Funktion der simulierten Umstimmereinrichtung in Abhängigkeit der Regelparameter nicht bekannt ist, führt dazu, dass nicht davon ausgegangen werden kann, dass diese stetig differenzierbar ist. Der neue Parametersatz liegt zwar in Gradientenrichtung, ist allerdings nicht besser als der Startpunkt.

Das Suchverfahren (Kapitel 3.4.2) wurde unter denselben Bedingungen getestet. Das Verfahren misst alle umliegenden Parametersätze um den Startpunkt und stellt den besten Wert als neuen Startpunkt ein. Das Ergebnis ist in Abbildung 11 zu sehen:

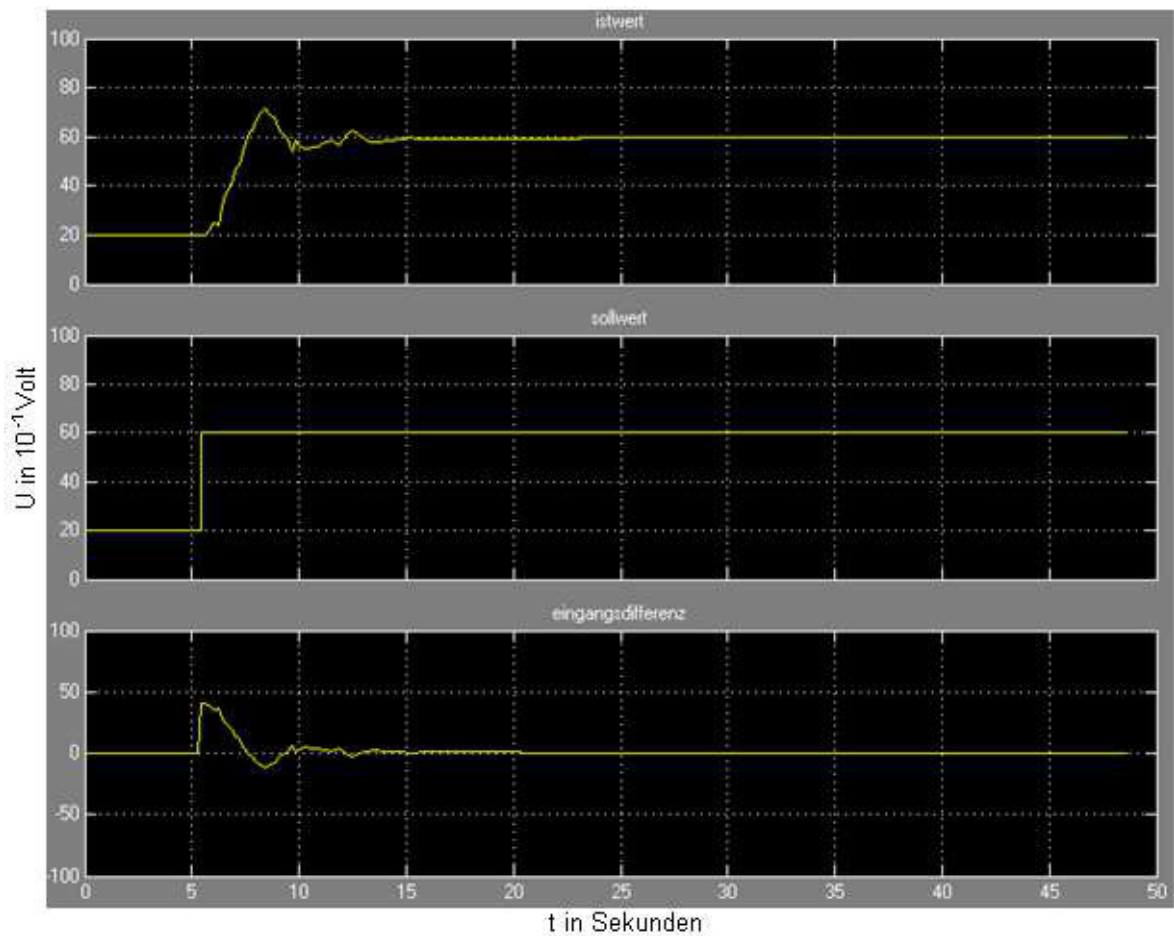


Abbildung 11: Simulationsergebnis der Optimierung mit Suchverfahren

Das Ergebnis ist deutlich besser, als das der Gradientenanalyse. Die Schwingungen wurden entfernt und die bleibende Regelabweichung ist minimiert. Allerdings dauert dieses Verfahren länger als das Gradientenverfahren.

In der Regelstrecke wurde nun eine Störgröße hinzugefügt. Es wurde ein Sinussignal mit einer Amplitude von 1 Volt und einer Frequenz von 0,1 Hz addiert. Die Frequenz wurde so gewählt, dass eine Periodendauer 10 Zeiteinheiten lang ist, daher $2\pi/10$. Das Ergebnis der Optimierung ist in Abbildung 12 zu sehen:

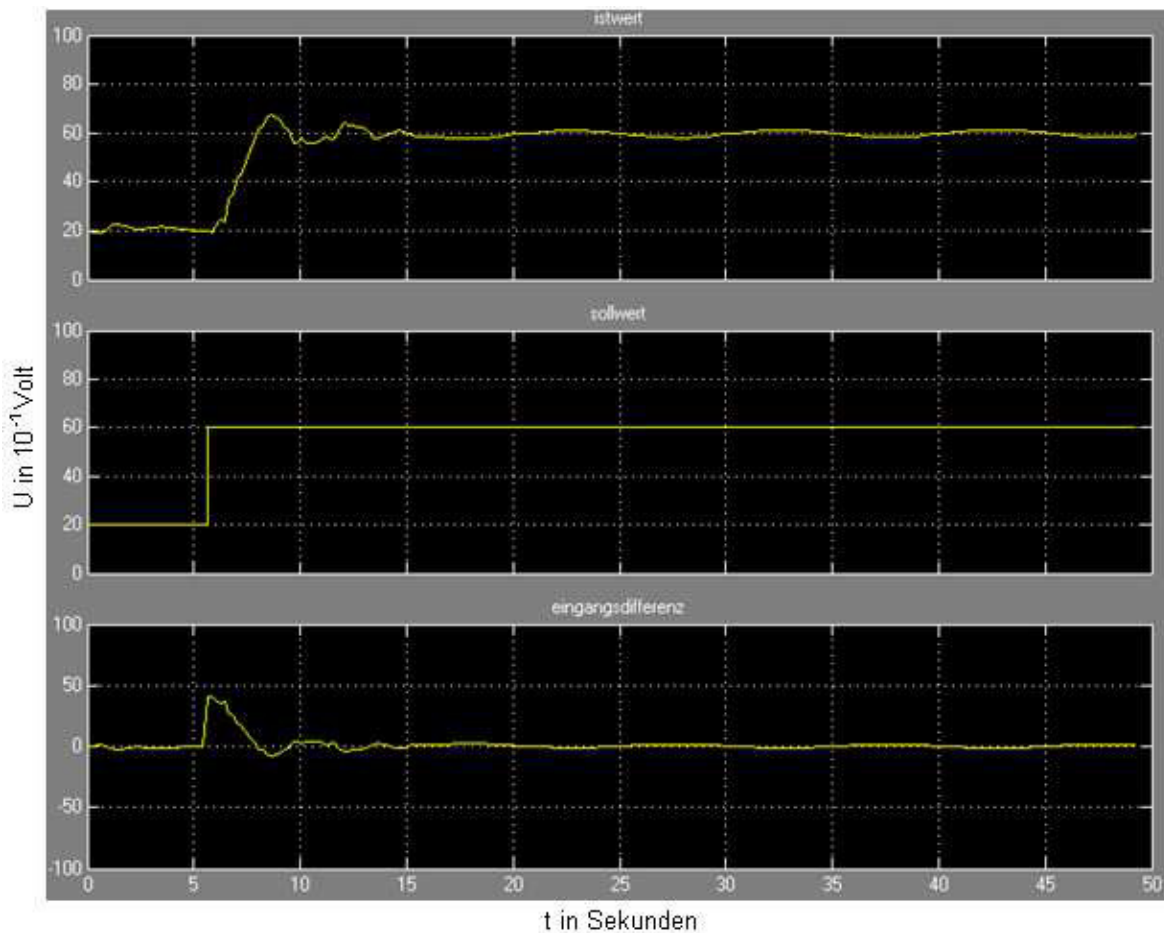


Abbildung 12: Simulationsergebnis mit Störgröße

Der Regler schafft mithilfe der neuen Parameter die Störgröße fast vollständig auszuregeln.

Zum Einbinden in die CAS-Firmware wurde das Suchverfahren (Kapitel 3.4.2.) gewählt. Dieses Verfahren wird im Weiteren als Optimierung bezeichnet.

4 Implementierung und Test

Nachdem sich in der Simulation das Suchverfahren als das Bessere herausstellte wurde dieses in die Programmiersprache C umgesetzt. Mittels einer CAS wurde das Verfahren an den realen Umstimmrichtungen getestet. Es soll zudem als Alternative zur automatischen Reglerparametrierung in die CAS-Firmware eingebunden werden. In Anlage B befinden sich Auszüge aus der Dokumentation zur CAS-Firmwareversion 1.4 [7].

4.1 Umsetzung des Optimierungsalgorithmus in C

Der Simulations-Algorithmus wurde in die Programmiersprache C umgesetzt, da die Firmware bereits in dieser Sprache implementiert ist. Dazu wurde zunächst die Qualitätserrechnung (Formel 2-14) der Reglerantwort implementiert. Dazu müssen 300 Messwerte, die im Abstand einer Millisekunde gemessen wurden, und der Sollwert genutzt werden. In der Berechnung werden nun alle Eingangsdifferenzen quadriert und aufsummiert. Anschließend werden, um die Ableitung mit einzubeziehen, die Differenzen der Messwerte gebildet. Diese werden ebenfalls quadriert und aufsummiert. Für die Gesamtsumme musste eine 64 Bit-Variable genutzt werden um einen Überlauf zu vermeiden. In Anlage B (Seite A-v) ist der endgültige Programmablauf der Qualitätserrechnung zu sehen (siehe Kapitel 4.3.1).

Zu Beginn müssen die Schrittweiten aufgestellt werden. Diese werden bei einer Messung der umliegenden Punkte zu den einzelnen Parametern addiert beziehungsweise subtrahiert. Als Schrittweiten wurden folgende Werte gewählt:

50	30	20	12	6	2	1
----	----	----	----	---	---	---

Zu beachten ist, dass der I- und der D-Anteil die Schrittweite in hundertstel betrachtet. Der P-Anteil betrachtet diese in Zehntel [5: S.15]. Die Schrittweiten wurden so gewählt, dass zu Beginn der Optimierung große Schritte (Grobparametrierung) gemacht werden und zum Ende kleine (Feinparametrierung). Die letzte Schrittweite sollte 1 betragen um das Minimum zu finden. Nun muss für jede Messung der Parametersatz eingestellt und ein Sollwert vorgegeben werden. Um den Parametersatz einzustellen wird ein Parameter mit der Schrittweite addiert beziehungsweise subtrahiert und die anderen werden auf die Startparameter gesetzt. Nach dem Sollwertsprung wird 300 Millisekunden lang im Regelsinterrupt, welcher aller Millisekunde ausgelöst wird, die Sensorspannung gemessen und in ein Array geschrieben. Dieses Array wird zusammen mit dem Sollwert an die Qualitätsfunktion übergeben, um die Güte des Parametersatzes zu messen. Nachdem alle Messungen durchgeführt wurden, wird der kleinste Wert ermittelt und die Regelparameter auf die zugehörigen Werte eingestellt. Im nächsten Durchlauf wird weder der Startwert noch der Wert, der im letzten Durchlauf der Startwert war, gemessen. Wurde beispielsweise im letzten Durchlauf der P-Anteil erhöht, so wird im nächsten Durchlauf die Mes-

sung mit verringertem P-Anteil übersprungen. Dazu wird der Qualitätswert der Messung auf den Maximalwert der 64 Bit-Variable gesetzt. Dieser Ablauf wiederholt sich so lange bis der Startwert der kleinste ist, denn dann wurde ein lokales Minimum gefunden. Die Schrittweite wird auf den nächsten Wert gesetzt und der Algorithmus beginnt von vorn. Nach dem letzten Schrittweitenwert ist der Algorithmus beendet. In Anlage B (Seite A-vi) befindet sich der Ablaufplan der Optimierung. Zudem befindet sich ein Quelltextauszug zu einer Messung eines Parametersatzes in Anlage B (Seite A-vii).

4.2 Einbindung in Firmware

4.2.1 Firmwareversion 1.4

In der Firmware existiert bereits die automatische Reglerparameterermittlung, die schneller ist als der Optimierungsalgorithmus, jedoch nicht immer zufriedenstellende Parameter errechnet. Da die Optimierung länger dauert, wurde diese zusätzlich in die Firmware eingebaut ohne die automatische Reglerparameterermittlung zu entfernen. Um die Optimierung in die Firmware einzubinden wurden sechs neue P-Codes¹ nach dem Vorbild der automatischen Reglerparameterermittlung in die folgende Revision des projektinternen CAN-Kommunikations-Protokoll eingeführt. Die P-Codes 26 bis 29 starten eine Optimierung an einer Umstimmeinrichtung. Als Startpunkt wird der aktuelle Parametersatz genutzt. Nach 25 Minuten wird eine laufende Optimierung abgebrochen, da nicht unendlich lange optimiert werden darf. Im Falle des Timeout-Abbruch werden die ermittelten Parameter nicht gespeichert, da nicht sichergestellt werden kann, dass diese Parameter ausreichend gut sind. Mittels P-Code 30 wird der Optimierungsstatus abgefragt. Die CAS antwortet dann mit einer von fünf möglichen Antworten, welche im ersten Byte der CAN-Nachricht mit dem Identifier 66 steht:

- 0: Die Optimierung ist noch nicht abgeschlossen
- 1: Die Optimierung ist abgeschlossen
- 2: Es wurde ein Sensor- oder Aktorfehler an der zu optimierenden Umstimmeinrichtung festgestellt, es findet keine Optimierung statt
- 3: Die Optimierung wurde nicht gestartet
- 4: Timeout, die Optimierung wurde nach 25 Minuten abgebrochen

Mittels dem P-Code 31 kann eine laufende Optimierung abgebrochen werden. In diesem Fall werden die Parameter auf die Werte eingestellt, die vor der Optimierung eingestellt waren.

¹ Ein P-Code ist ein Parameter-Code der zur Kommunikation zwischen ZAS und CAS genutzt wird, siehe [5]: S.10f

4.2.2 Firmwareversion 1.50

Zum Testen wurde eine Firmwarevariante 1.50 [8] entworfen. Diese wurde für Tests entworfen und basiert auf Version 1.3. In dieser Firmwareversion wurde die automatische Reglerparameterermittlung durch die Optimierung ersetzt. Zudem wurden zwei neue P-Codes eingebunden, welche in der endgültigen Firmware nicht vorhanden sein werden. P-Code 26 ermöglicht einen Sollwertsprungtest, mit welchem man die Güte der Regelparame-ter feststellen kann. Im 6. Byte muss die Umstimmeinrichtungsnummer stehen, welche getestet werden soll. Innerhalb des Sollwertsprungtests werden nun nacheinander 16 Sollwerte im Abstand von 300 Millisekunden vorgegeben. Durch diesen Sollwert-sprungtest soll das gesamte Verhalten des Reglers zu sehen sein. Dieser Vorgang dauert 4,8 Sekunden. Zudem wurde ein P-Code 27 eingebunden, woraufhin die CAS mit einem CAN-Telegramm antwortet, in dem die Anzahl der Durchläufe für die einzelnen Schrittwei-ten der letzten Optimierung steht. Diese Daten sollten dazu dienen, überflüssige Schritt-weiten zu identifizieren und zu entfernen, um die Gesamtoptimierungszeit zu verkürzen. Für diese Firmware wurde auch die Regelungsfunktion verändert. Beim Optimieren der Umstimmeinrichtungen wird die Sollwertvorgabe an einem anderen Stellgrößenausgang ausgegeben [8]. Dies ist notwendig, um während der Optimierung bereits die Güte der Parameter zu sehen. Durch die Ausgabe des Sollwertes kann (durch Nutzung eines Oszil-loskops) dieser direkt mit dem Istwert verglichen werden, um Regelabweichungen zu er-kennen. Zudem lassen sich Einschwingzeiten messen.

4.3 Tests und Ergebnisse

Um die Parameter zu testen wurden verschiedene Sollwertvorgaben genutzt. Die Abbil-dungen 13 - 43 zeigen Aufzeichnungen der Istwertverläufe bei vorgegebenen Sollwerten. Zudem wird die Differenz Istwert - Sollwert dargestellt.

4.3.1 Labortest

Eine vorhandene Umstimmeinrichtung wurde mit dem Algorithmus getestet. Bei der ge-testeten Umstimmeinrichtung handelt es sich um eine kleine, leichte Regelstrecke. Ge-testet wurde die dritte Umstimmeinrichtung (Kapitel 2.2). Sie besitzt wenig Reibung und bewegt wenig Masse. Innerhalb des Algorithmus wird ein Sollwertsprung zur Qualitätsbe-stimmung genutzt. Als Testsprung wurde ein Sprung von 1,5 Volt auf 3,0 Volt gewählt. In Abbildung 13 ist die resultierende Reglerantwort auf den Sollwertsprung zu sehen.

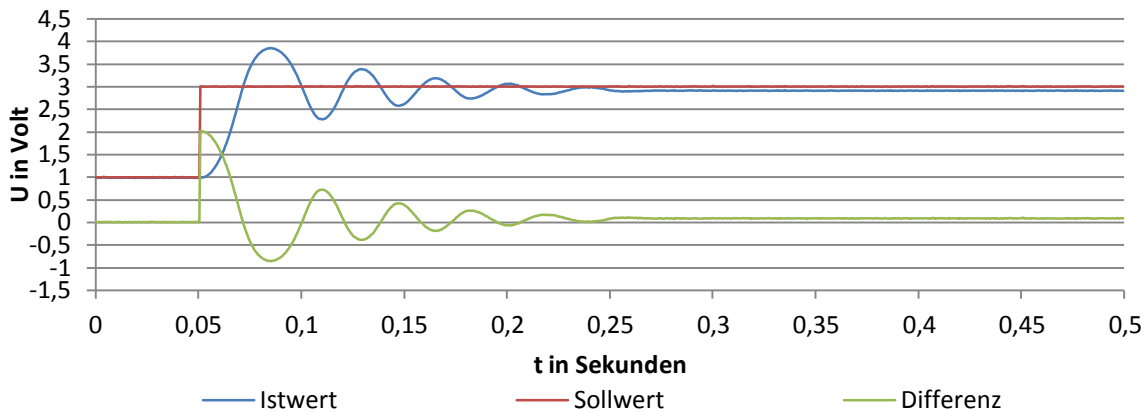


Abbildung 13: Reglerantwort nach Optimierung mit einem Sprung

Allerdings ist nun nur der getestete Sprung optimiert, das übrige Reglerverhalten ist nicht optimal. Abbildung 14 zeigt einen Test mit einer Reihe von Sollwerten nach einer Optimierung mit dem Suchverfahren:

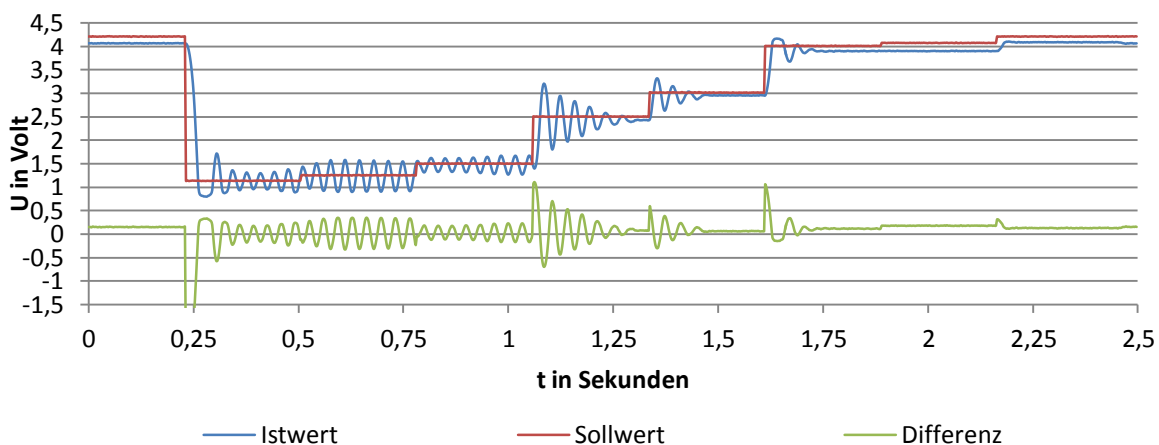


Abbildung 14: Sollwertvorgaben nach Optimierung mit einem Sprung

Zu sehen ist ein deutliches Schwingverhalten für Sollwertvorgaben unter 2,5 Volt. Zudem ist im Kennlinienbereich oberhalb von 3,5 Volt eine Abweichung zu erkennen. Wird in der Optimierung nur ein Sprung genutzt, wird dieser Sprung zwar optimiert, alle anderen jedoch nicht. Daher müssen mehrere Sprünge zur Qualitätsberechnung genutzt werden. Die verwendeten Sprünge müssen den gesamten Sensorbereich abdecken und große sowie kleine Sprünge beinhalten um das gesamte Reglerverhalten zu erfassen. Für jeden Sprung muss eine Qualitätsberechnung (Kapitel 2.3.3) stattfinden, deren Werte summiert werden. Von einer Mittelwertbildung wurde abgesehen, da einerseits im Microcontroller die Division einer 64-Bit-Zahl sehr zeitaufwändig wäre und es zudem mathematisch keinen Unterschied macht, ob der Mittelwert oder die Gesamtsumme von gleich vielen Werten verglichen wird. Als Sollwerte wurden folgende Werte gewählt:

1,0 Volt 1,6 Volt 2,2 Volt 2,8 Volt 3,4 Volt 4,0 Volt 2,5 Volt

Die Optimierung mit sieben Sollwerten wurde durch verschiedene Sollwertvorgaben getestet. Das Ergebnis ist in Abbildung 15 zu sehen:

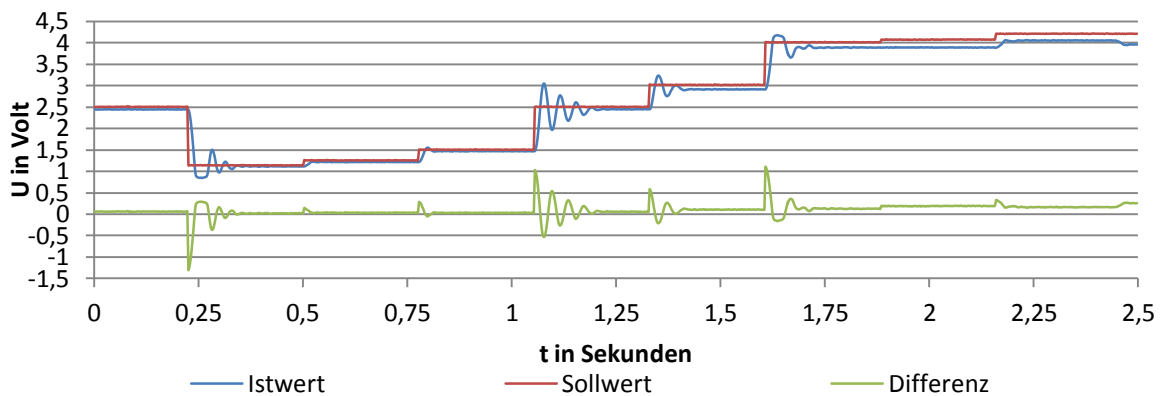


Abbildung 15: Sollwertvorgaben nach Optimierung mit 7 Sprüngen

Das Reglerverhalten hat sich gegenüber der Optimierung mit einem Sprung deutlich verbessert, ist aber noch nicht zufriedenstellend, da nach wie vor Abweichungen zu sehen sind. Daher wurden mehr Sollwerte zur Optimierung betrachtet. Die Sollwerte wurden so gewählt, dass zu Beginn ein Stufensprung von 1,0 Volt bis 4,0 Volt in 0,3-Volt-Schritten durchgeführt wird, damit der gesamte nutzbare Kennlinienbereich getestet wird. Anschließend werden zwei Sprünge mit einer Distanz von 1,5 Volt ausgeführt damit das Verhalten bei großen Sprüngen ebenso mit einbezogen wird. Abschließend wird ein Stufensprung von 4,0 Volt auf 2,5 Volt durchgeführt. Abbildung 16 zeigt das Ergebnis der Optimierung durch Vorgabe von Sollwerten:

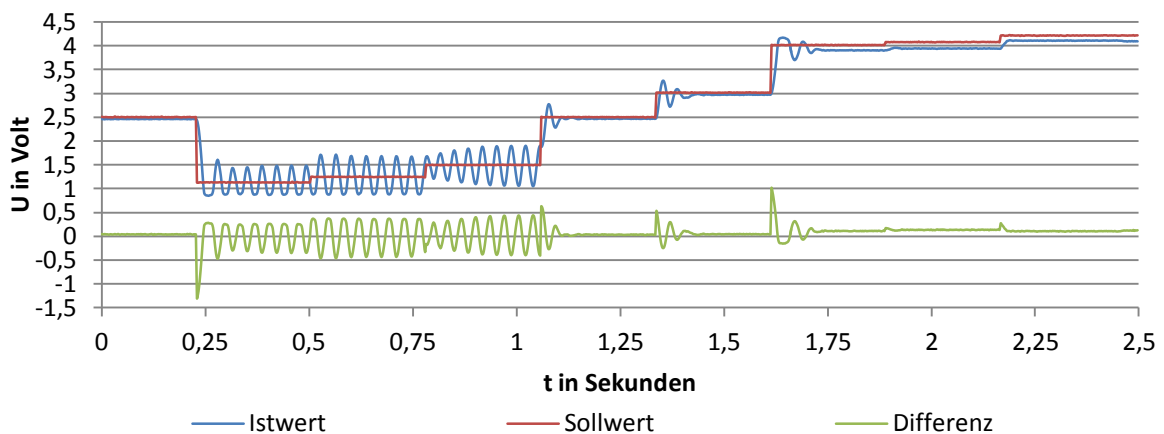


Abbildung 16: 8 Sollwertvorgaben nach Optimierung mit 16 Sprüngen

Ebenso wurden die Sollwertsprünge des Optimierungsverfahrens zur Gütebetrachtung genutzt. Dies ist in Abbildung 17 zu sehen:

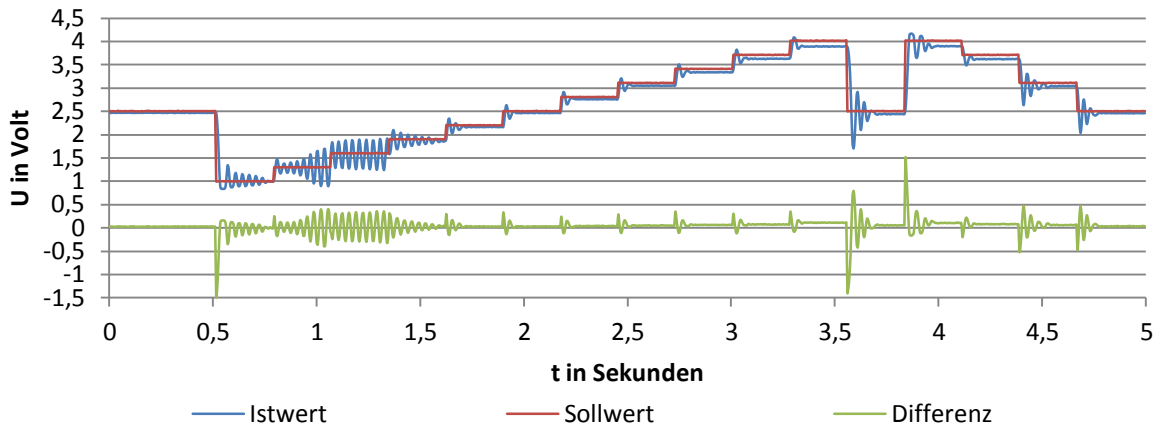


Abbildung 17: 16 Sollwertvorgaben nach Optimierung mit 16 Sprüngen

Zu sehen ist eine Schwingung im Kennlinienbereich unter 2 Volt sowie leichte Abweichungen zwischen Soll- und Istwert. Trotzdem wurde dieses Reglerverhalten von dem Algorithmus als optimal errechnet. Da der Algorithmus die Güte nur anhand des Qualitätsmerkmals betrachtet, musste dieses geändert werden. Da Schwingverhalten ungünstig ist, muss dieses im Qualitätsmerkmal gewichtet werden. Daher wurde der Ableitungsteil mit Faktor 50 multipliziert, damit ein schwingendes Reglerverhalten einen deutlich schlechteren Qualitätswert besitzt als ein ruhiger. Aus den letzten zehn quadratischen Eingangsdifferenzen wurde der Mittelwert gebildet und mit dem Gesamtergebnis multipliziert, um eine bleibende Regelabweichung im Qualitätsmerkmal feststellen zu können. Zu beachten ist, dass der Mittelwert mit eins addiert werden muss, da der Mittelwert null sein kann, damit wäre der Qualitätswert ebenso null. Da innerhalb der ersten 50 Millisekunden Schwingverhalten normal ist und dieses Schwingen von der Sprungdistanz abhängig ist, wurde nun nur noch das Reglerverhalten nach 50 Millisekunden betrachtet. Das neue Qualitätsmerkmal errechnet sich nach Änderung der Formel 2-13 folgendermaßen:

$$Q = \left(\int_{50}^{\infty} ((e(\tau))^2) d\tau + 50 * \int_{50}^{\infty} \left(\left(\frac{de(\tau)}{d\tau} \right)^2 \right) d\tau \right) * \left(\left(\frac{1}{10} * \int_{\infty-10}^{\infty} ((e(\tau))^2) d\tau \right) + 1 \right) \quad (4 - 1)$$

Für die Implementierung wurde die Formel diskretisiert. Die unendliche Messzeit wurde mit 300 Millisekunden ersetzt:

$$Q = \left(\sum_{t=50}^{300} [(e(t))^2] + 50 * \sum_{t=50}^{300} [(e(t) - e(t-1))^2] \right) * \left(\left(\frac{1}{10} * \sum_{t=290}^{300} [(e(t))^2] \right) + 1 \right) \quad (4 - 2)$$

$$e(t) = \text{Sollwert} - \text{Istwert}(t)$$

Der Programmablaufplan der Qualitätserrechnung ist in Anlage B (Seite A-v) hinterlegt. Dieses Qualitätsmerkmal ist nur für Regelstrecken anwendbar, welche innerhalb von 300 Millisekunden einen eingeschwungenen Zustand erreichen können. Für langsamere Regelstrecken ist dieses Merkmal nicht zu benutzen, da es zu keiner optimalen Lösung kommen kann. Für Regelstrecken, welche den Einschwingvorgang in deutlich unter 50 Millisekunden beendet haben, ist dieses Qualitätsmerkmal ebenfalls nicht anwendbar. Das neue Qualitätsmerkmal wurde ebenfalls nach einer Optimierung mit einer Reihe Sollwerten getestet. Die Ergebnisse sind in Abbildung 18 und 19 zu sehen:

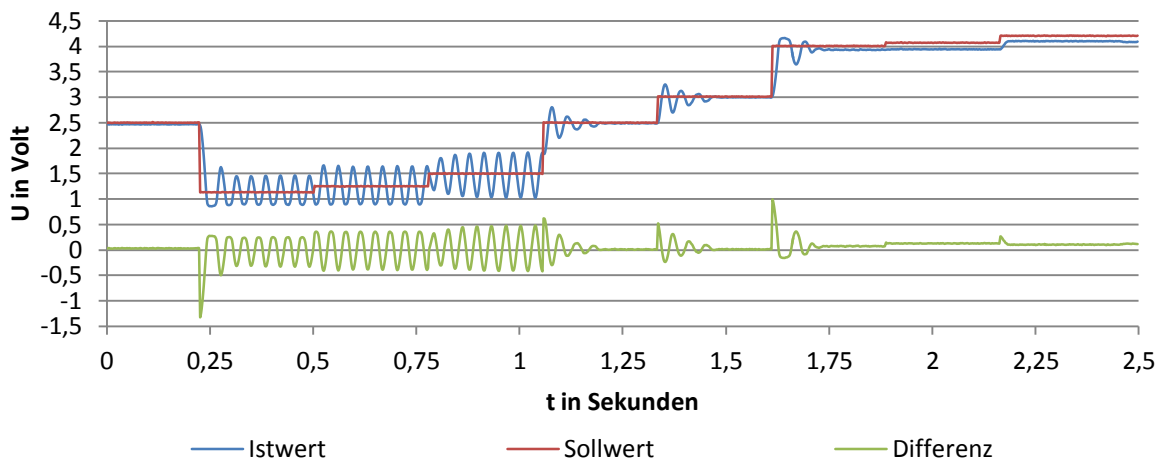


Abbildung 18: 8 Sollwertvorgaben nach Optimierung mit neuem Qualitätsmerkmal

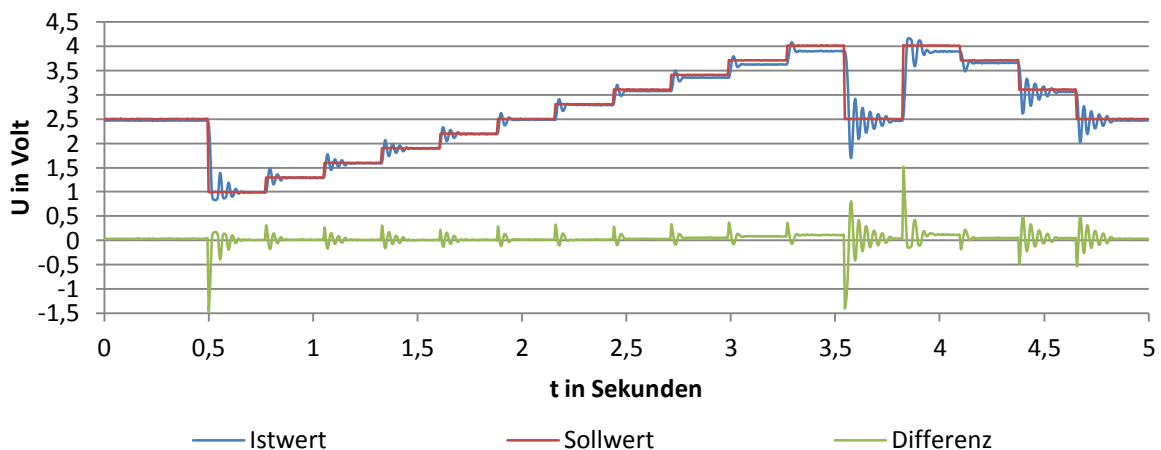


Abbildung 19: 16 Sollwertvorgaben nach Optimierung mit neuem Qualitätsmerkmal

Durch das neue Qualitätsmerkmal werden die Ergebnisse etwas besser, jedoch ist in Abbildung 18 noch immer ein Schwingverhalten im Kennlinienbereich unter 2,5 Volt zu sehen. Laut „Spezifikation der Kommunikation über den CAN-BUS“ Revision 5 [5: S.15] werden die PID-Parameter unterschiedlich an den Regelalgorithmus weitergegeben. Der P-Anteil wird in Zehnteln verarbeitet, während I- und D-Anteil in Hundertstel verarbeitet werden. Der jetzige Optimierungsalgorithmus behandelt die Anteile allerdings gleich, was dazu führt, dass der P-Anteil im Gegensatz zu den anderen Anteilen zehnfach verstärkt

wird. Da der Regelalgorithmus nicht geändert werden darf, da dies Änderungen in der ZAS sowie in den Konfigurationsdateien nach sich ziehen würde, wurde die Behandlung des P-Anteils im Optimierungsalgorithmus geändert. Für die Messungen mit höherem P-Anteil wird nun nur noch ein Zehntel der Schrittweite zum P-Anteil addiert. Bei der Messung mit verringertem P-Anteil wird nur noch ein Zehntel der Schrittweite subtrahiert. Dadurch werden die Parameter gleich gewertet. Die Ergebnisse nach einer weiteren Optimierung sind in Abbildung 20 und 21 zu sehen:

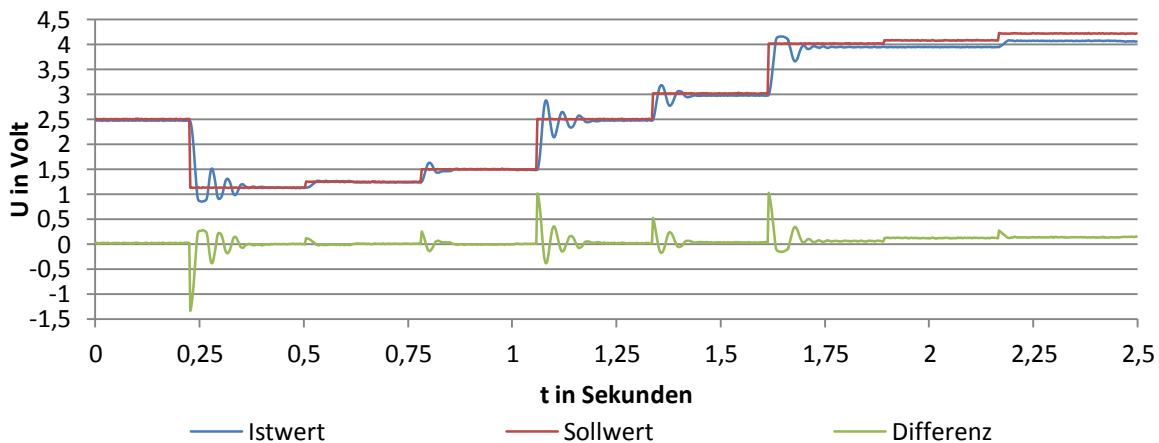


Abbildung 20: 8 Sollwertvorgaben nach Optimierung mit geteiltem P-Anteil

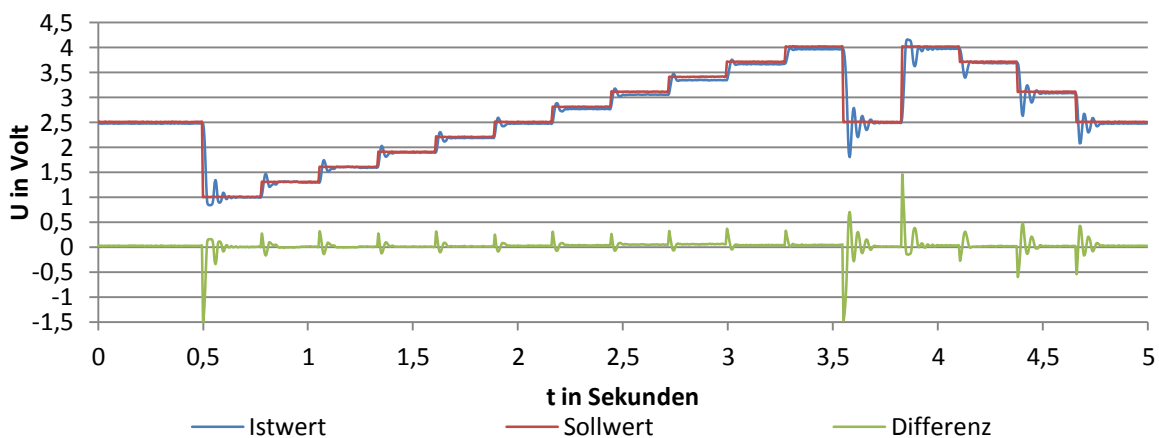


Abbildung 21: 16 Sollwertvorgaben nach Optimierung mit geteiltem P-Anteil

Die Ergebnisse der Optimierung sind zufriedenstellend. Das Schwingverhalten wurde weitestgehend entfernt und die bleibende Regelabweichung ebenso. An den Grenzwerten sind noch Regelabweichungen zu beobachten, was an den mechanischen Grenzen der Umstimmereinrichtung liegt. Da der Algorithmus jedoch für alle vorhandenen Regelstrecken anwendbar sein muss, wurden auch andere Regelstrecken getestet. Getestet wurde nun eine Regelstrecke mit mehr Reibung. Die erste Umstimmereinrichtung (Kapitel 2.2) wurde genutzt. Die Ergebnisse der Optimierung sind in den Abbildungen 22 und 23 zu sehen:

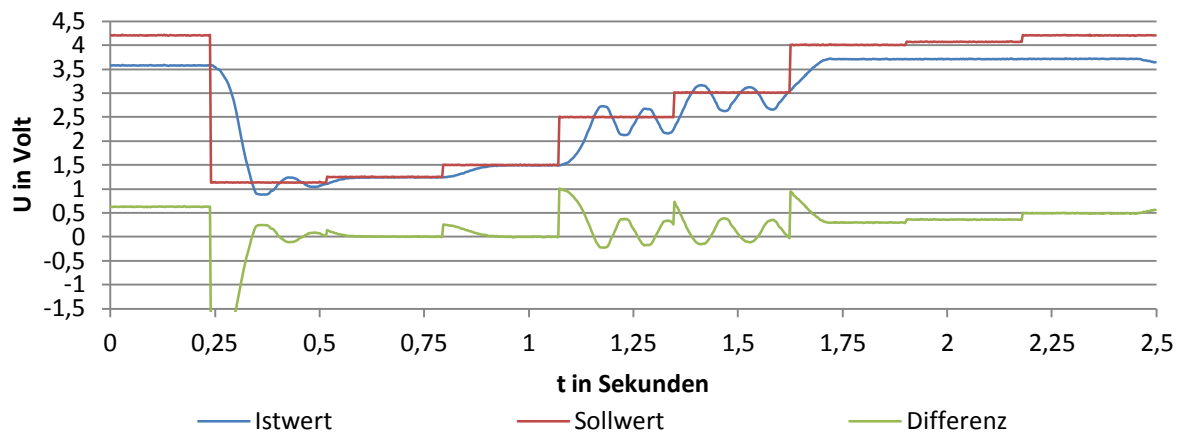


Abbildung 22: 8 Sollwertvorgaben nach Optimierung

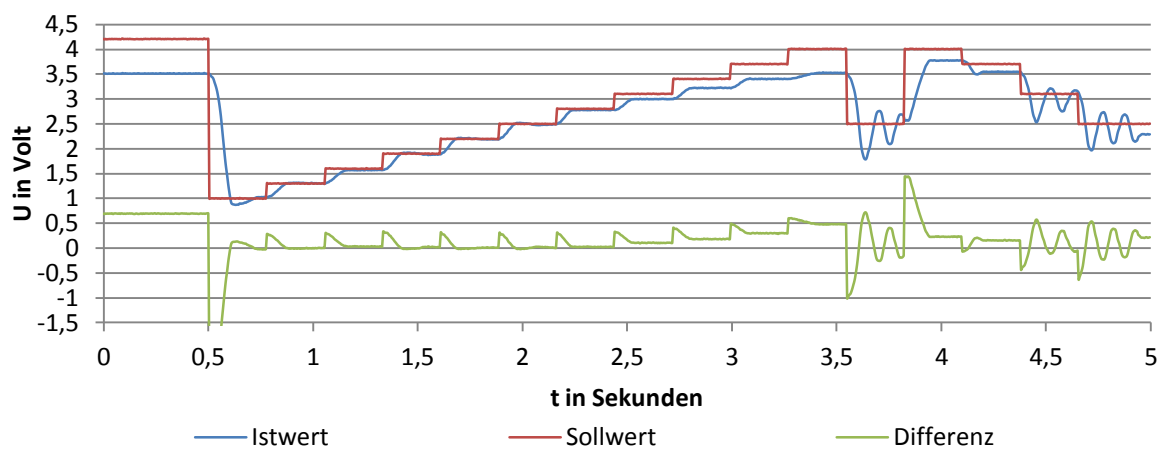


Abbildung 23: 16 Sollwertvorgaben nach Optimierung

Die Ergebnisse sind nicht zufriedenstellend. Zu sehen ist eine zu große bleibende Abweichung sowie Schwingverhalten. Da diese Umstimmereinrichtung sehr viel Reibung aufweist, ist ein hoher P-Anteil nötig. Daher wurde die Teilung des P-Anteils wieder entfernt und eine neue Parametrierung durchgeführt. Die Ergebnisse sind in den Abbildungen 24 und 25 zu sehen:

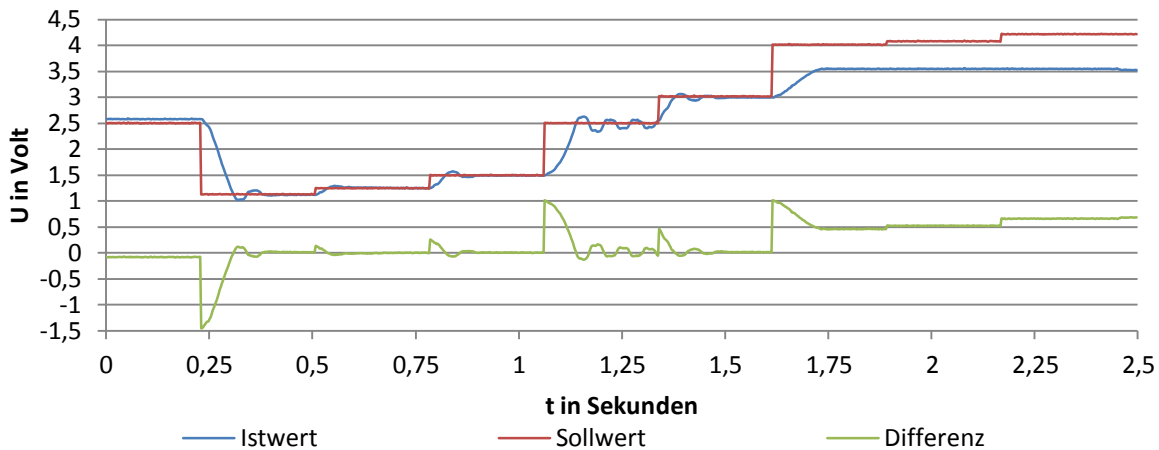


Abbildung 24: 8 Sollwertvorgaben nach Optimierung

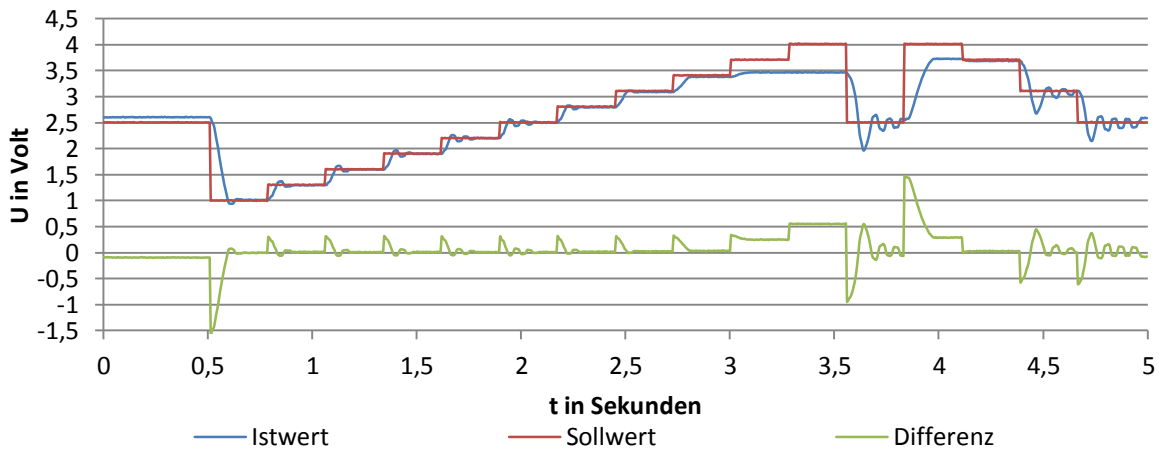


Abbildung 25: 16 Sollwertvorgaben nach Optimierung

Das Schwingverhalten sowie die bleibende Abweichung wurden deutlich reduziert. Da allerdings kleine Umstimmrichtungen eine Teilung des P-Anteil benötigen, muss vor der Optimierung ein Testsprung durchgeführt werden, um festzustellen, welcher Art die angeschlossene Umstimmrichtung angehört. Hierzu wird die Aktorspannung auf ein Volt eingestellt und nach 300 Millisekunden der Sensorwert gemessen. Anschließend wird der Aktor auf vier Volt eingestellt und nach weiteren 300 Millisekunden die Aktorspannung gemessen. Falls die Differenz der Sensorwerte größer ist als 2,5 Volt, so liegt eine kleine, leichte Umstimmrichtung vor und der P-Anteil wird geteilt. Andernfalls liegt eine große, schwere Umstimmrichtung vor und der P-Anteil wird nicht geteilt.

Der Algorithmus ist in der Lage einen vorhandenen Parametersatz als Startpunkt zu nutzen. Bislang wurden alle Tests mit einem Nullpunktstart durchgeführt, dass bedeutet alle Parameter sind zu Beginn null. Anhand einer schnellen Umstimmrichtung wurde der parametrierte Start nach einer automatischen Reglerparameterermittlung getestet. Ebenfalls soll getestet werden, ob die Verarbeitungszeit des Algorithmus gesenkt werden kann, da dieser mittlerweile über zehn Minuten benötigt. Zuerst wurde der Optimierungsalgo-

rithmus mit dem Startpunkt null getestet. Die Optimierungszeit betrug 16 Minuten. Das Ergebnis ist in Abbildung 26 zu sehen:

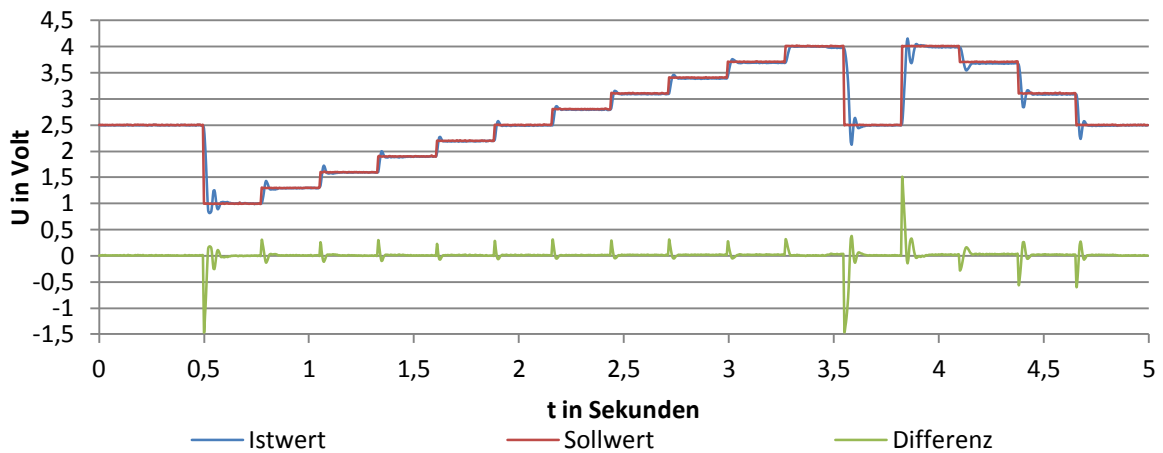


Abbildung 26: Sollwertvorgaben nach Optimierung mit Nullpunktstart

Die PID-Parameter wurden genullt und eine automatische Reglerparameterermittlung (ARE) gestartet. Anschließend wurde eine Optimierung durchgeführt. Das Ergebnis ist in Abbildung 27 zu sehen:

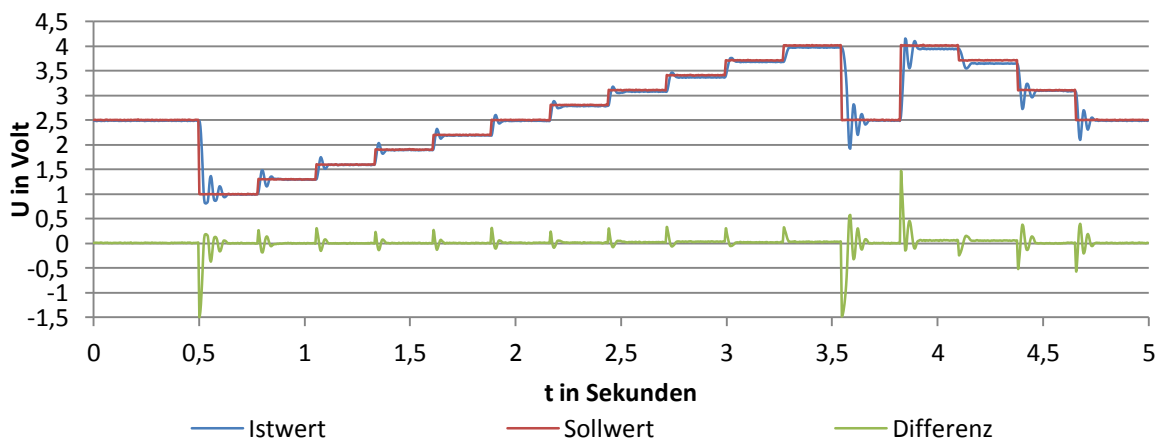


Abbildung 27: Sollwertvorgaben nach ARE und Optimierung

Nach dem parametrierten Start ist das Reglerverhalten ebenso gut wie nach dem Nullpunktstart. Die Verarbeitungszeit nach dem parametriertem Start beträgt jedoch nur sieben Minuten, was eine deutliche Verbesserung gegenüber dem Nullpunktstart ist.

4.3.2 Gegenüberstellung der Verfahren

Bei der Gegenüberstellung werden für alle vier Umstimmrichtungen die automatische Reglerparameterermittlung, die Optimierung mit parametriertem Start sowie die Nullpunktstart-Optimierung verglichen. Die Umstimmrichtungen eins und zwei sind hierbei große Umstimmrichtungen mit viel Reibung und daher träge. Die Umstimmrichtungen

gen drei und vier sind klein und leicht (Kapitel 2.2). In Tabelle 1 sind die Verarbeitungszeiten zu sehen:

Umstimm- element	Automatische Reglerpa- rameterermittlung	Optimierung nach Paramet- rierung (ARE)	Optimierung (Nullpunktstart)
1	70 Sekunden	10 Minuten	8 Minuten
2	70 Sekunden	10 Minuten	4 Minuten
3	50 Sekunden	6 Minuten	16 Minuten
4	40 Sekunden	8 Minuten	19 Minuten

Tabelle 1: Parametrierungszeiten

Zu sehen ist, dass schwere Umstimmrichtungen nach einem Nullpunktstart schneller Parametrieren werden. Leichte Umstimmrichtungen jedoch werden nach einem Parametrisierten Start schneller Parametrieren.

4.3.2.1 Umstimmrichtung 1

Die Abbildungen 28, 29 und 30 zeigen das Reglerverhalten nach den einzelnen Verfahren. Tabelle 2 zeigt die errechneten Reglerparameter:

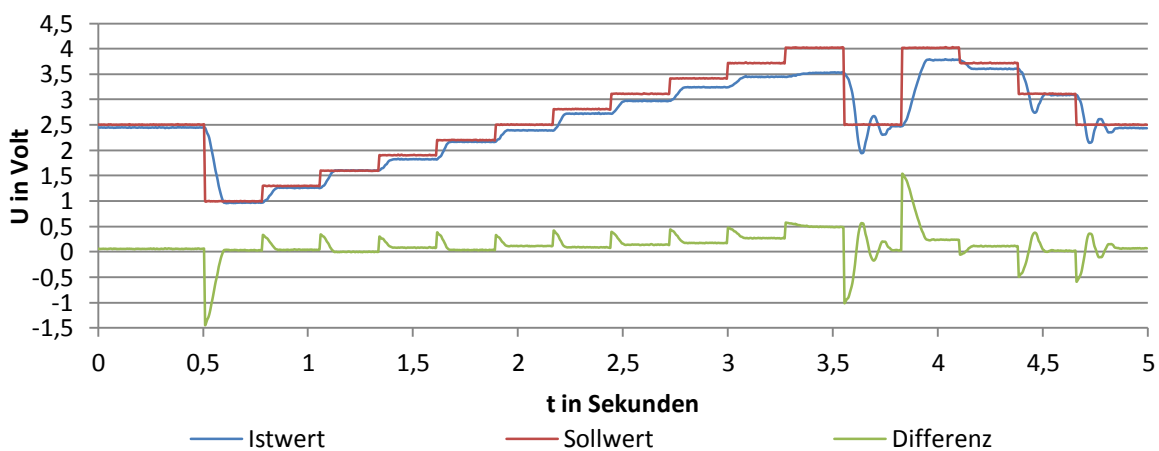


Abbildung 28: Umstimmrichtung 1 Automatische Reglerparameterermittlung

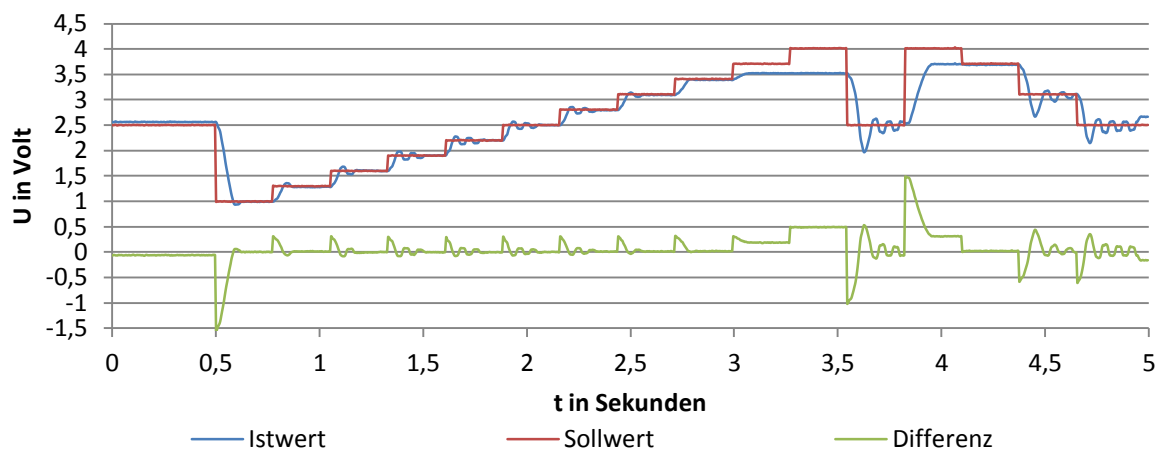


Abbildung 29: Umstimmeinrichtung 1 Optimierung nach ARE

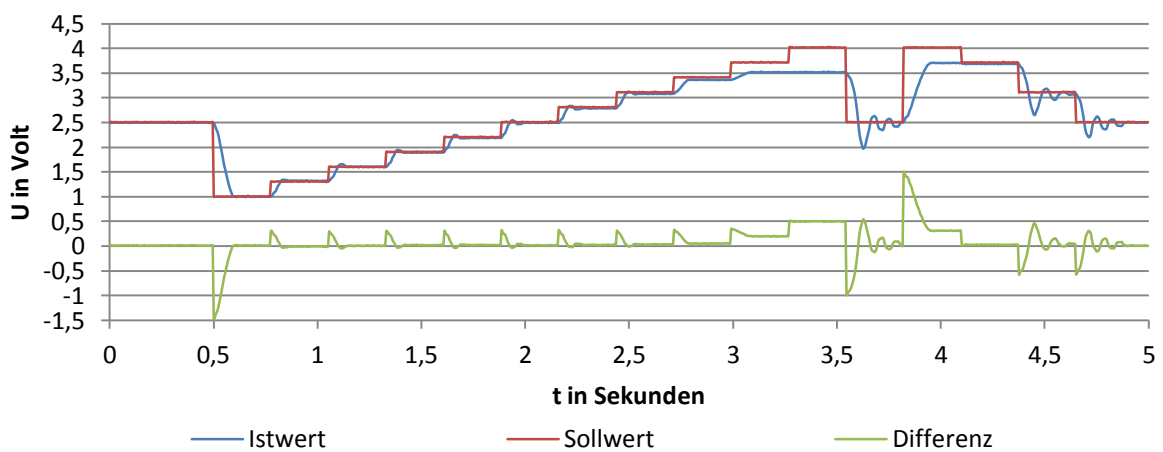


Abbildung 30: Umstimmeinrichtung 1 Nullpunktstart-Optimierung

	P-Anteil	I-Anteil	D-Anteil
Automatische Reglerparameterermittlung	5,4	0,4	6,98
Parametrierte Optimierung	23,6	0,64	5,82
Nullpunktstart-Optimierung	15,0	0,42	0

Tabelle 2: Umstimmeinrichtung 1 Reglerparameter

Bei dieser Umstimmeinrichtung wurde das beste Ergebnis nach der Nullpunktstart-Optimierung erzielt. Die Verfahrbereichsgrenze von 3,5 Volt ist mechanisch bedingt. Eine bleibende Regelabweichung ist nicht mehr zu sehen. Das Schwingverhalten ist minimiert.

4.3.2.2 Umstimmereinrichtung 2

In den Abbildungen 31, 32 und 33 sind das Reglerverhalten nach den Verfahren zu sehen. Die errechneten Reglerparameter sind in Tabelle 3 verzeichnet:

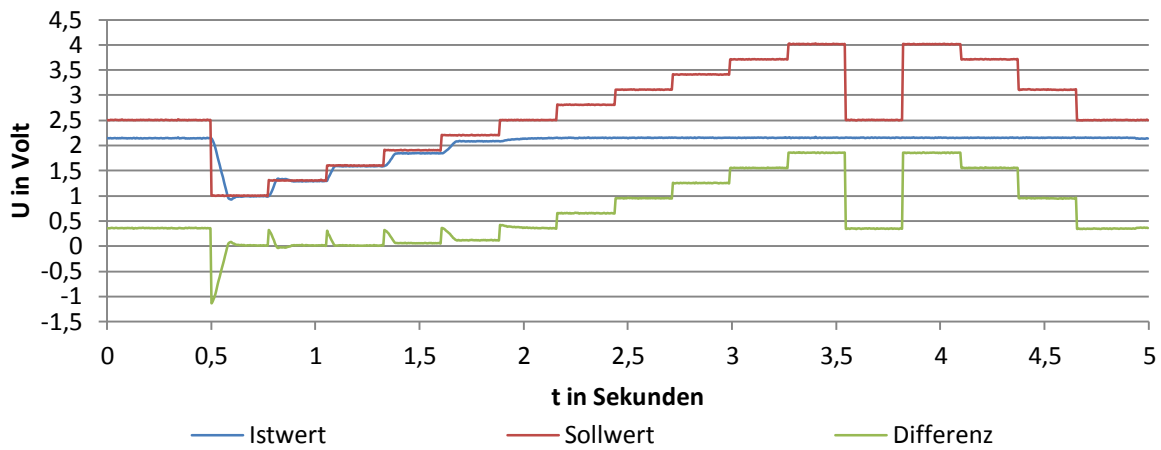


Abbildung 31: Umstimmereinrichtung 2 Automatische Reglerparameterermittlung

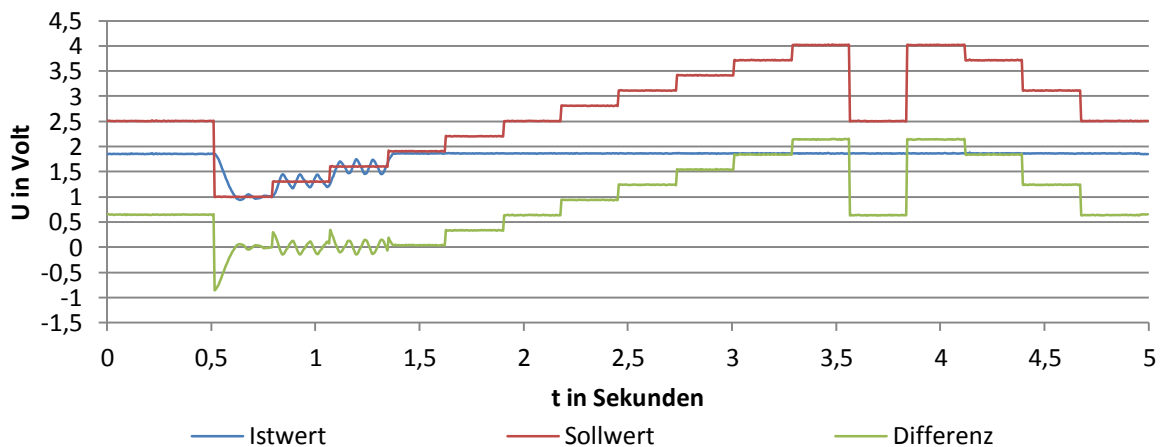


Abbildung 32: Umstimmereinrichtung 2 Optimierung nach ARE

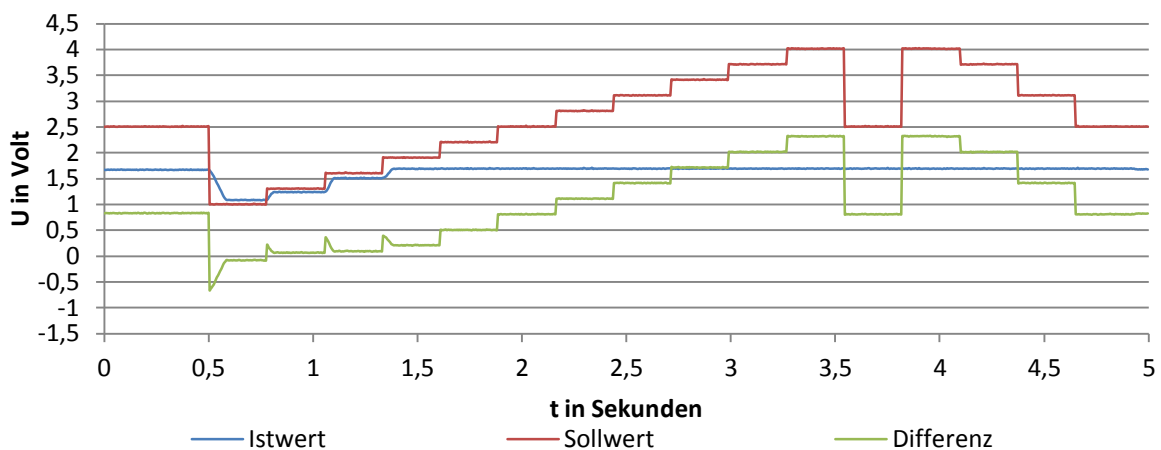


Abbildung 33: Umstimmeinrichtung 2 Nullpunktstart-Optimierung

	P-Anteil	I-Anteil	D-Anteil
Automatische Reglerparameterermittlung	9,8	0,76	2,17
Parametrierte Optimierung	7,4	2,82	2,11
Nullpunktstart-Optimierung	9,8	0,02	0

Tabelle 3: Umstimmeinrichtung 2 Reglerparameter

Die Ergebnisse der Umstimmeinrichtung 2 sind nicht optimal. Die Spulen des Tauchspulenantriebs können nicht genügend Kraft aufbringen, um das Umstimmelement ausreichend zu bewegen. Die besten Ergebnisse erzielte hier die automatische Reglerparameterermittlung, da mit diesen Parametern der untere Kennlinienbereich bis 2 Volt genutzt werden kann.

4.3.2.3 Umstimmeinrichtung 3

In den Abbildungen 34, 35 und 36 ist das Reglerverhalten nach den Verfahren zu sehen. Die Parameter sind in Tabelle 4 zu sehen:

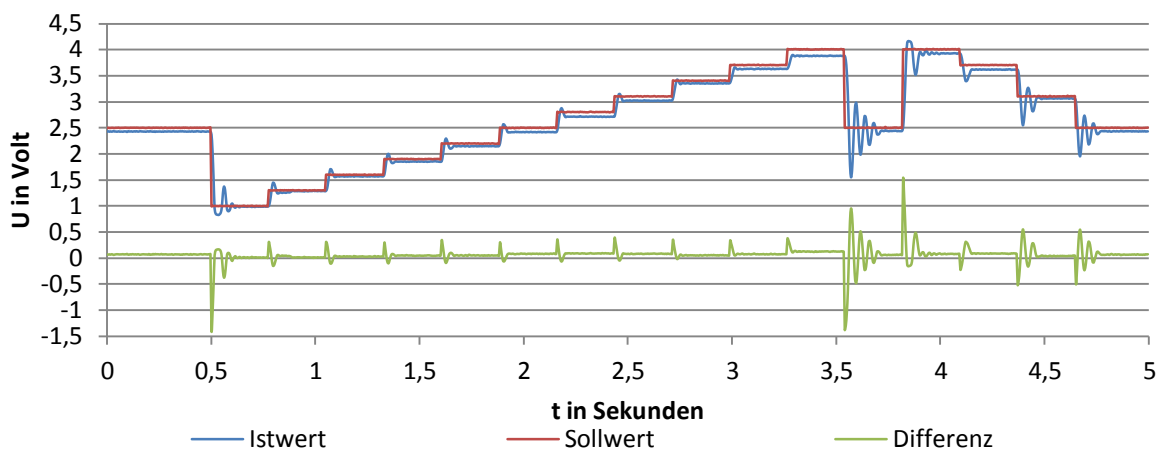


Abbildung 34: Umstimmeinrichtung 3 Automatische Reglerparameterermittlung

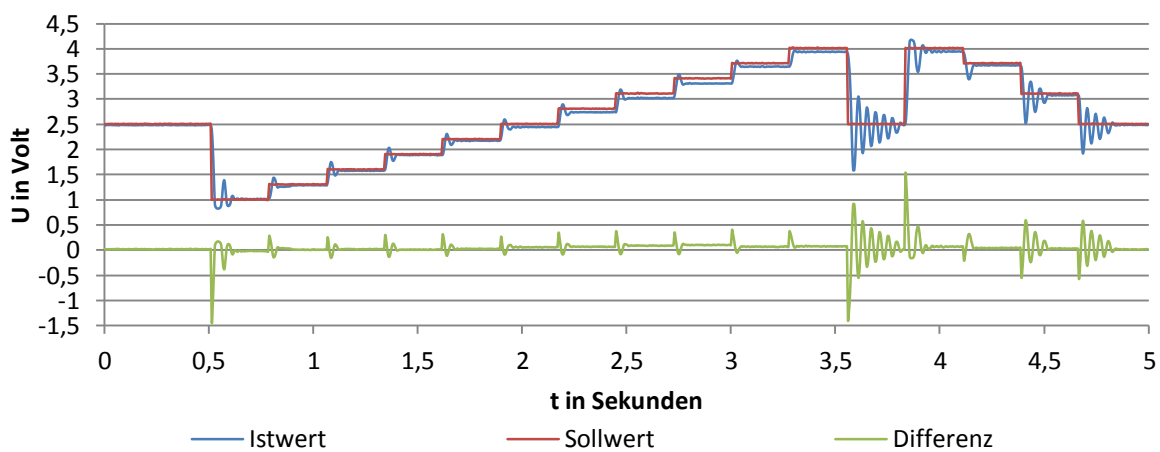


Abbildung 35: Umstimmeinrichtung 3 Optimierung nach ARE

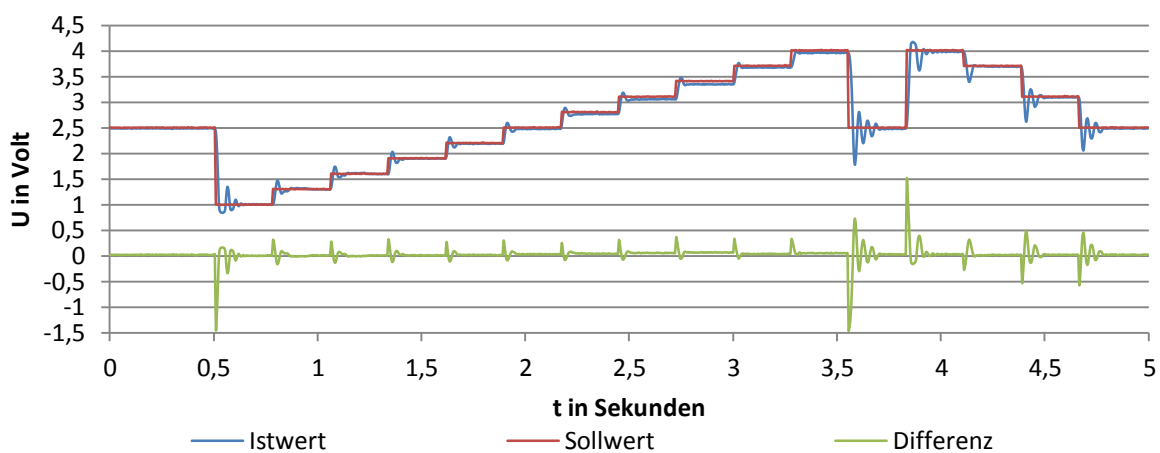


Abbildung 36: Umstimmeinrichtung 3 Nullpunktstart-Optimierung

	P-Anteil	I-Anteil	D-Anteil
Automatische Reglerparameterermittlung	2,7	0,16	0,85
Parametrierte Optimierung	3,2	0,18	0,87
Nullpunktstart-Optimierung	3,4	0,3	7,04

Tabelle 4: Umstimmeinrichtung 3 Reglerparameter

Das beste Ergebnis wurde nach dem Nullpunktstart erzielt, auch wenn die Verarbeitungszeit am längsten war. Zu sehen ist ebenfalls, dass sich die Parameter nicht viel unterscheiden. Das zeigt, dass die Minimalpunkte der Funktion dieser Regelstrecke nah beieinander liegen. Im Vergleich mit anderen Regelstrecken wird deutlich, dass dies nicht die Regel ist.

4.3.2.4 Umstimmeinrichtung 4

In den Abbildungen 37, 38 und 39 sind die Ergebnisse der Verfahren für Umstimmeinrichtung 4 zu sehen. Tabelle 5 zeigt die Regelparameter:

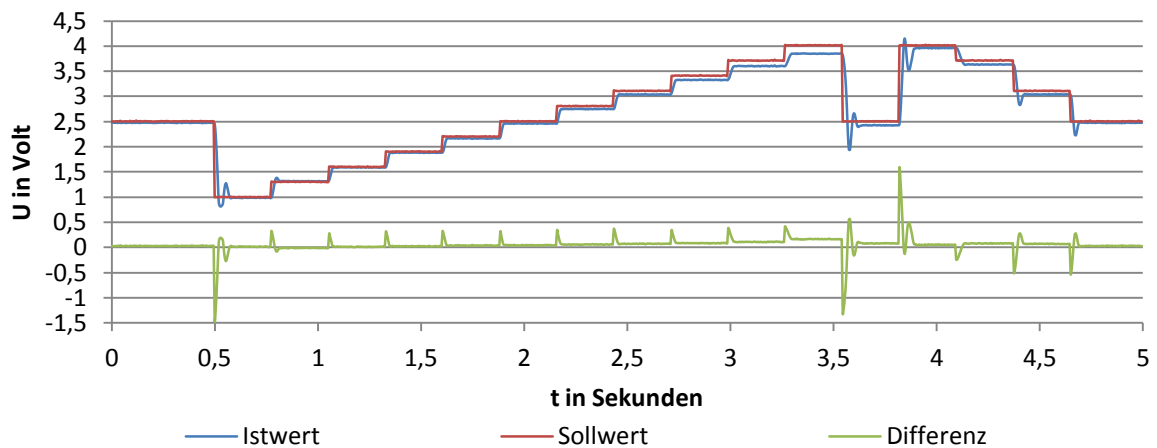


Abbildung 37: Umstimmeinrichtung 4 Automatische Reglerparameterermittlung

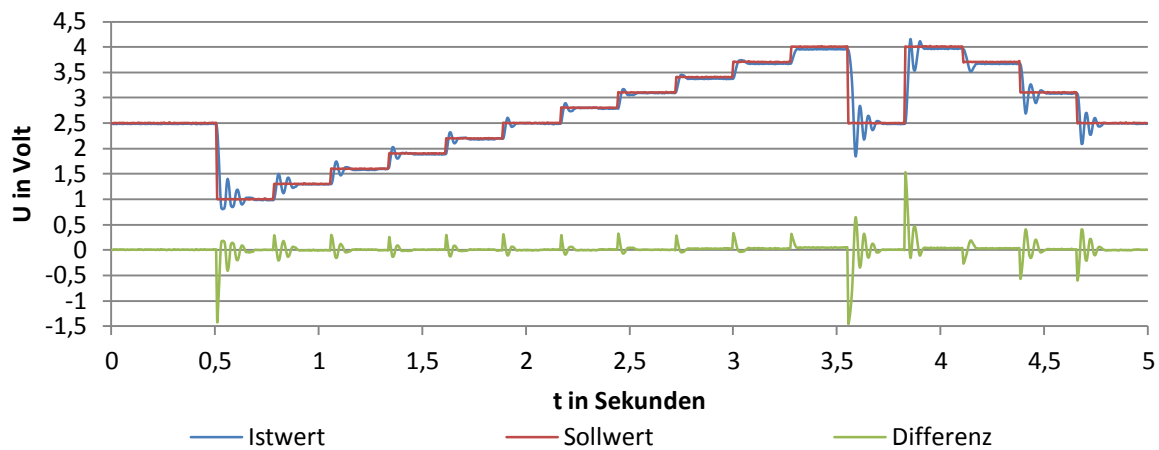


Abbildung 38: Umstimmeinrichtung 4 Optimierung nach ARE

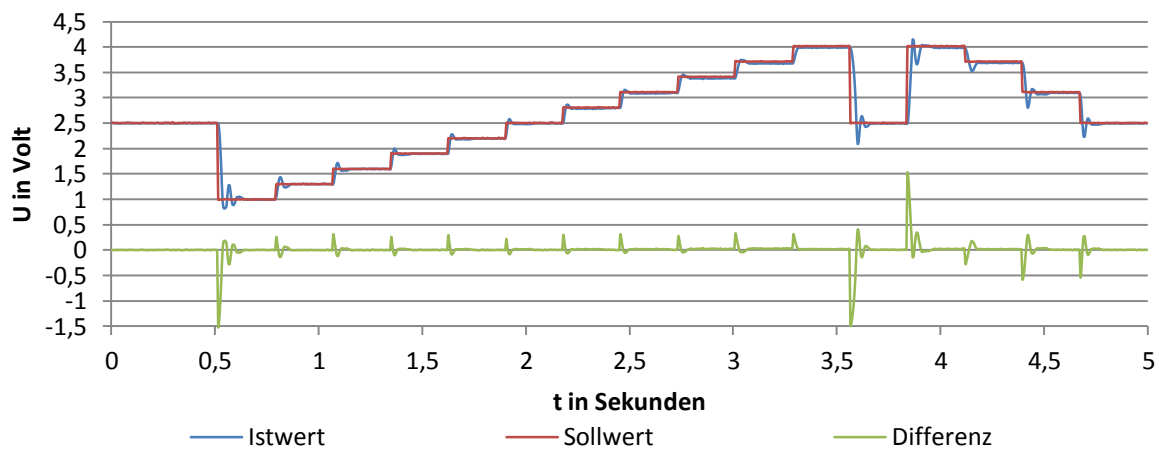


Abbildung 39: Umstimmeinrichtung 4 Nullpunktstart-Optimierung

	P-Anteil	I-Anteil	D-Anteil
Automatische Reglerparameterermittlung	2,7	0,06	0,73
Parametrierte Optimierung	2,9	0,3	1,4
Nullpunktstart-Optimierung	3,7	0,35	10,57

Tabelle 5: Umstimmeinrichtung 4 Reglerparameter

Bei dieser Umstimmeinrichtung wurde das beste Ergebnis nach dem Nullpunktstart erzielt. Die errechneten Parameter sind sich sehr ähnlich, was bedeutet, dass durch die einzelnen Parametrierungsverfahren das annähernd gleiche Minimum gefunden wurde.

4.3.3 Feldtest

Der Algorithmus wurde während eines Feldtests an der realen Orgel getestet. Während des Feldtest wurden alle Umstimmereinrichtungen mittels der automatischen Reglerparameterermittlung eingestellt. Bei zwei Umstimmereinrichtungen, deren Ergebnisse nicht zufriedenstellend waren, wurde der Optimierungsalgorithmus angewandt. Dazu wurde eine CAS mit der Firmwareversion 1.50 verwendet. Zum Darstellen des Reglerverhaltens wurde der Sollwertsprungtest verwendet. Abbildung 40 zeigt das Reglerverhalten der ersten Umstimmereinrichtung nach der automatischen Reglerparameterermittlung.

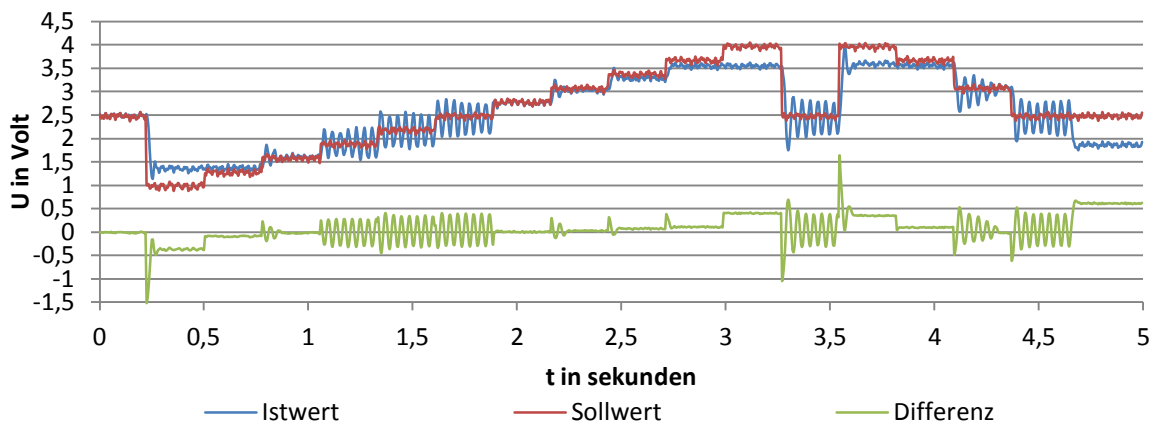


Abbildung 40: Sollwertsprungtest Umstimmenelement 1 Automatische Reglerparameterermittlung

Es ist zu sehen, dass der Istwert bei Sollwerten um 2,5 Volt schwingt. Dieses Verhalten ist ungünstig, da dem Ton eine zusätzliche Schwingung überlagert wird. Die Abweichung an den Grenzbereichen 1,5 Volt und 3,5 Volt ist auf den mechanischen Aufbau zurückzuführen. Um das Verhalten zu verbessern wurde der Optimierungsalgorithmus als Nullpunktstart angewandt. Das Ergebnis der Optimierung ist in Abbildung 41 zu sehen.

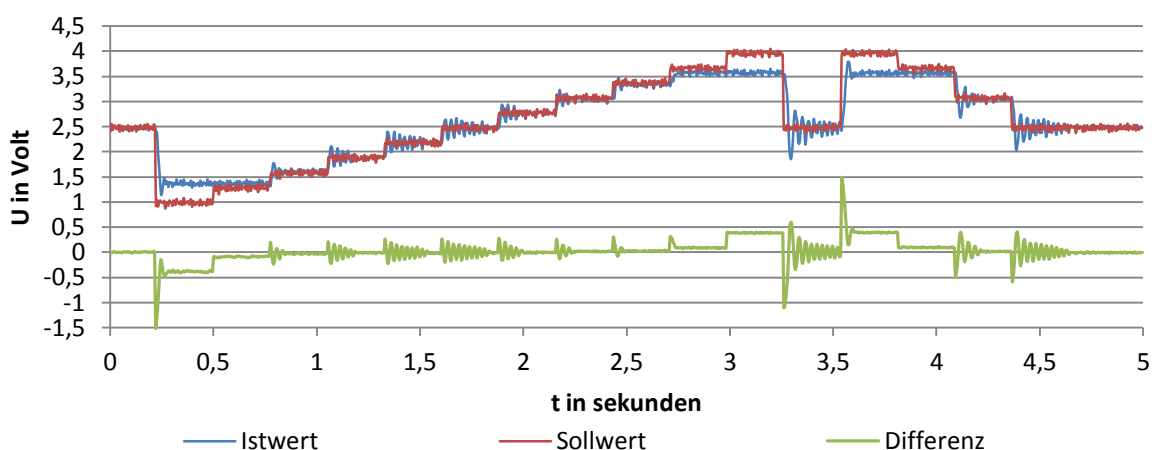


Abbildung 41: Sollwertsprungtest Umstimmenelement 1 Optimierung

Das Schwingverhalten im Bereich 2,5 Volt wurde zwar deutlich verringert, konnte allerdings nicht vollständig entfernt werden. Ein leichtes Überspringen ist noch zu beobachten. Die Optimierungszeit betrug 6 Minuten.

Bei der zweiten Umstimmereinrichtung waren nach der automatischen Reglerparameterermittlung ebenfalls Schwingungen zu beobachten. Dies ist in Abbildung 42 zu sehen.

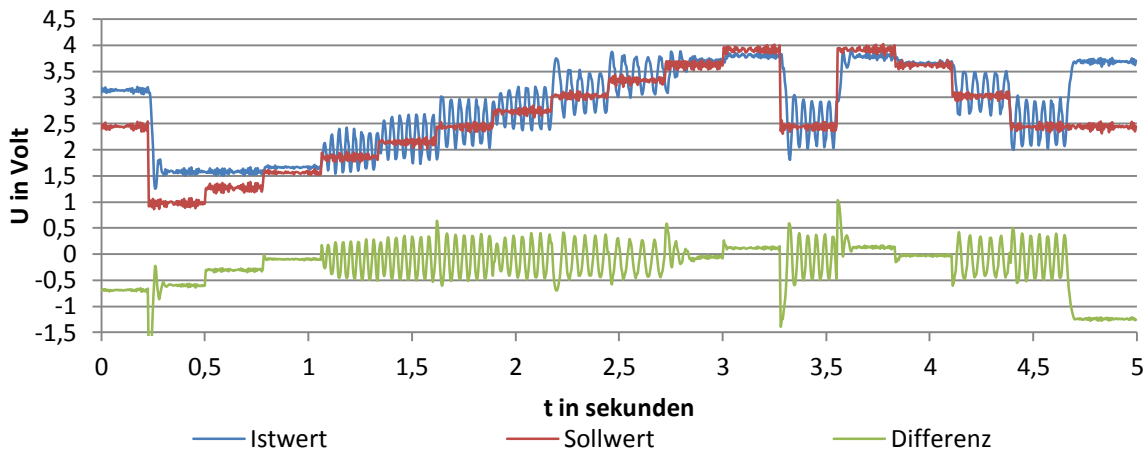


Abbildung 42: Sollwertsprungtest Umstimmelement 2 Automatische Reglerparameterermittlung

Bei dieser Umstimmereinrichtung wurde ebenfalls der Optimierungsalgorithmus angewandt. Es wurde eine Nullpunktstart-Optimierung durchgeführt. Das Ergebnis ist in Abbildung 43 zu sehen.

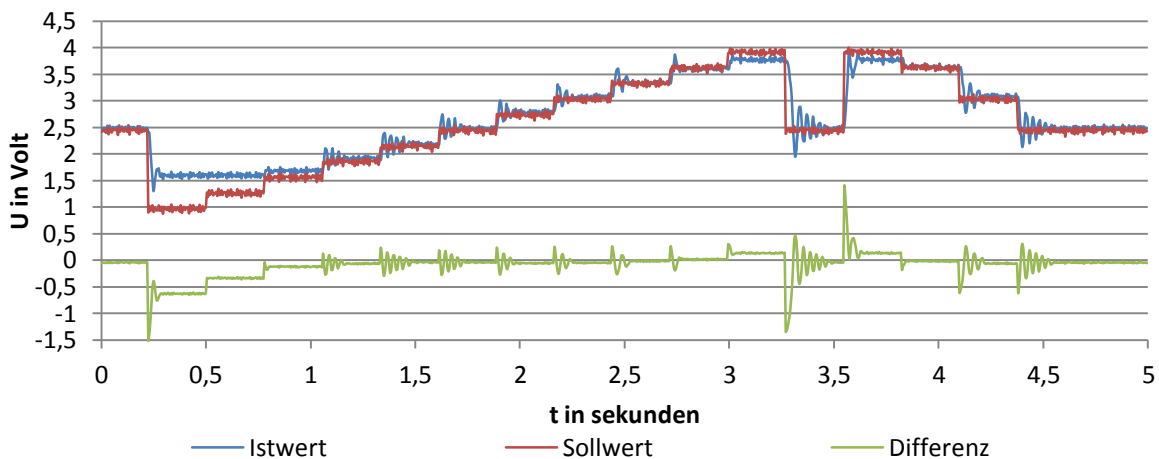


Abbildung 43: Sollwertsprungtest Umstimmelement 2 Optimierung

Die Schwingungen wurden weitestgehend entfernt. Das Reglerverhalten hat sich deutlich gegenüber den Ergebnissen der automatischen Reglerparameterermittlung verbessert. Bei dieser Umstimmereinrichtung ist ebenso eine mechanische Grenze bei Sollwerten unter 1,5 Volt zu erkennen. Die Optimierungszeit betrug 8 Minuten.

5 Fazit

5.1 Ergebnisse

Das Optimierungsverfahren arbeitet zuverlässig, da bei allen getesteten Regelstrecken optimale Parameter gefunden wurden. Dadurch dass das Verfahren alle möglichen Parameteränderungen probiert, ist es nicht relevant, ob es sich bei der angeschlossenen Umstimmeinrichtung um eine große oder kleine handelt. Eine bleibende Regelabweichung oder Schwingverhalten sind nicht mehr zu beobachten. Der Algorithmus braucht zwar mehr Zeit als eine automatische Reglerparameterermittlung, liefert allerdings bessere Ergebnisse ab. Das Verfahren ließe sich ebenso auf andere Regelstrecken anwenden, sofern diese eine geringe Schwingung aushalten, ohne beschädigt zu werden.

Nach wie vor kann allerdings nicht garantiert werden, dass der Sollwert innerhalb der geforderten 30 Millisekunden erreicht wird. Diese Zeit hängt nicht nur von den Regelparametern ab, sondern auch von den physikalischen Eigenschaften des Systems und der Sprungdistanz, der Soll- Istwertdifferenz. Für die im Labor vorhandenen Umstimmeinrichtungen eins und zwei kann nicht garantiert werden, dass diese in unter 30 Millisekunden eingeschungen sind, da die Reibung zu groß ist. Umstimmeinrichtung drei und vier sind jedoch schnell genug.

5.2 Ausblick

Das Verfahren könnte weiterentwickelt werden, beispielsweise indem der Algorithmus solange einen Parameter erhöht, bis keine Verbesserung mehr zu messen ist und dann erst alle Richtungen prüft. Das würde Zeit sparen. Ebenso ist es denkbar, das Downhill-Simplex-Verfahren zur Optimierung zu nutzen. Das Problem hierbei ist nur, dass das Verfahren, wie das Gradientenverfahren, auf einen unbekannten Parametersatz springt und das Verhalten an diesem Punkt unbekannt ist, da die mathematische Funktion der Regelstrecke unbekannt und nicht differenzierbar ist. Möglich wäre auch ein Neuentwurf der Umstimmeinrichtungen mit Spindelmotorantrieb, da diese präziser zu steuern sind. Allerdings sind diese Antriebe auch teurer als Tauchspulenantriebe. Die CAS-Platine könnte mit einem leistungsstärkeren Microcontroller ausgestattet werden, um beispielsweise einen Modellbasierten Regelalgorithmus zu verwenden.

Literatur

- [1] Thormann, Christian: „Entwicklung einer Firmware für Client-Aktor-Steuerungen zur dynamischen Umstimmung von Orgelpfeifen“, Mittweida, Labor Embedded Control, 2012
- [2] Unbehauen, Heinz: „Regelungstechnik I“, Braunschweig/Wiesbaden, Vieweg Verlag, 2002
- [3] Isermann, Rolf: „Digitale Regelsysteme: Band I Grundlagen deterministischer Regelungen“, Berlin/Heidelberg, Springer Verlag, 1988
- [4] Alt, Walter: Nichtlineare Optimierung, Braunschweig/Wiesbaden, Vieweg Verlag, 2002
- [5] Färber, Jan: „Spezifikation der Kommunikation über den CAN-BUS“, Mittweida, Labor Embedded Control, 2012, Revision 5
- [6] Thormann, Christian: „Dokumentation zur Simulation Regleroptimierung“, Mittweida, Labor Embedded Control, 2013
- [7] Thormann, Christian: „Dokumentation zur CAS-Firmware-Erweiterung 1.4“, Mittweida, Labor Embedded Control, 2013
- [8] Thormann, Christian: „Dokumentation zur Erweiterung FWV 1.50“, Mittweida, Labor Embedded Control, 2013

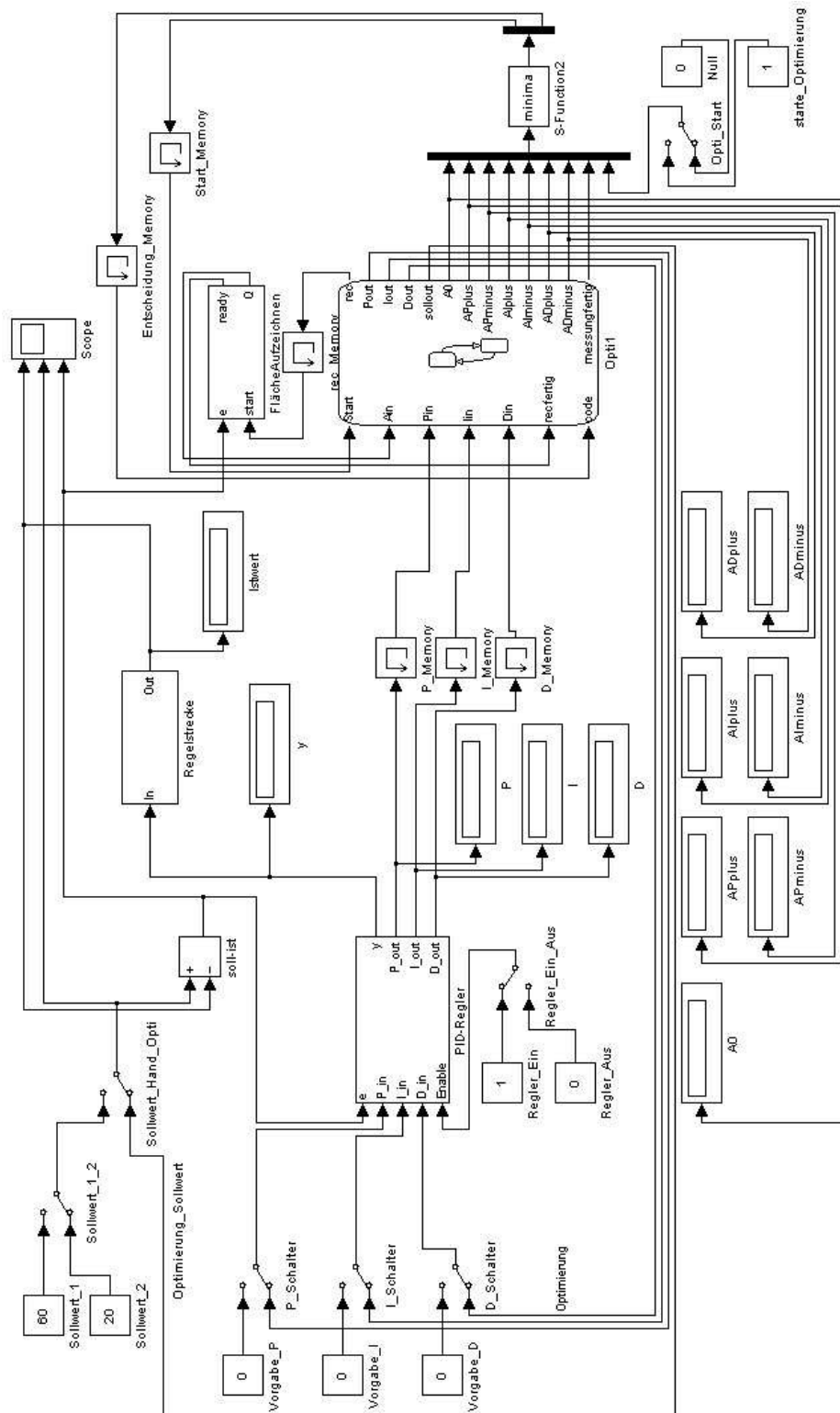
Anlagenverzeichnis

A: Dokumentation zur Simulation Regleroptimierung.....	A-i
A1: Gesamtsimulation.....	A-i
A2: Stateflow Optimierung.....	A-iii
B: Dokumentation zur CAS-Firmware-Erweiterung 1.4.....	A-v
B1: Ablaufplan Qualitätserrechnung.....	A-v
B2: Ablaufplan Optimierung.....	A-vi
B3: Quelltextauszug zur Messung eines Parametersatzes.....	A-vii

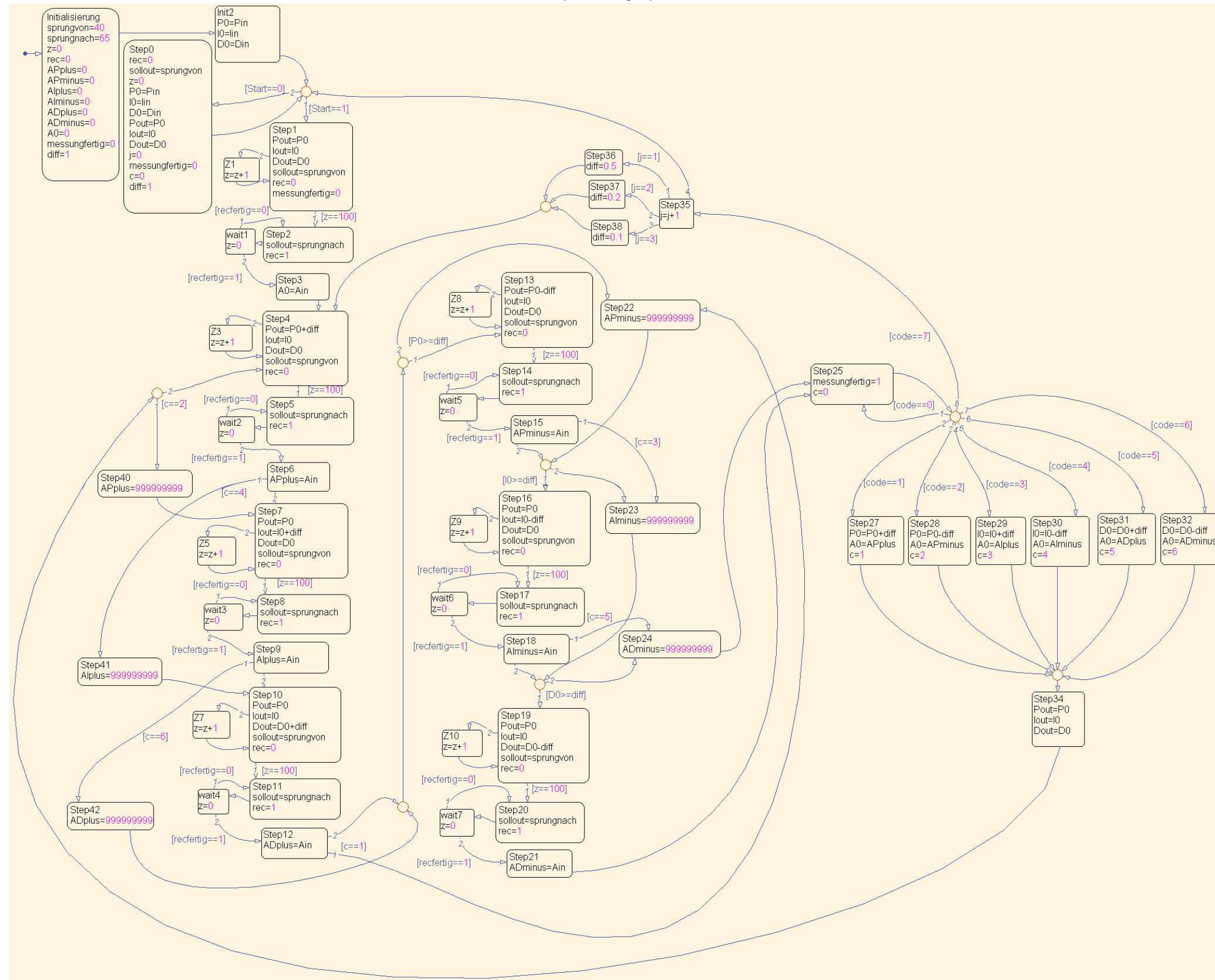
A: Dokumentation zur Simulation Regleroptimierung

Die Gesamte Dokumentation zur Simulation [6] befindet sich auf der beigelegten CD.

A1: Gesamtsimulation



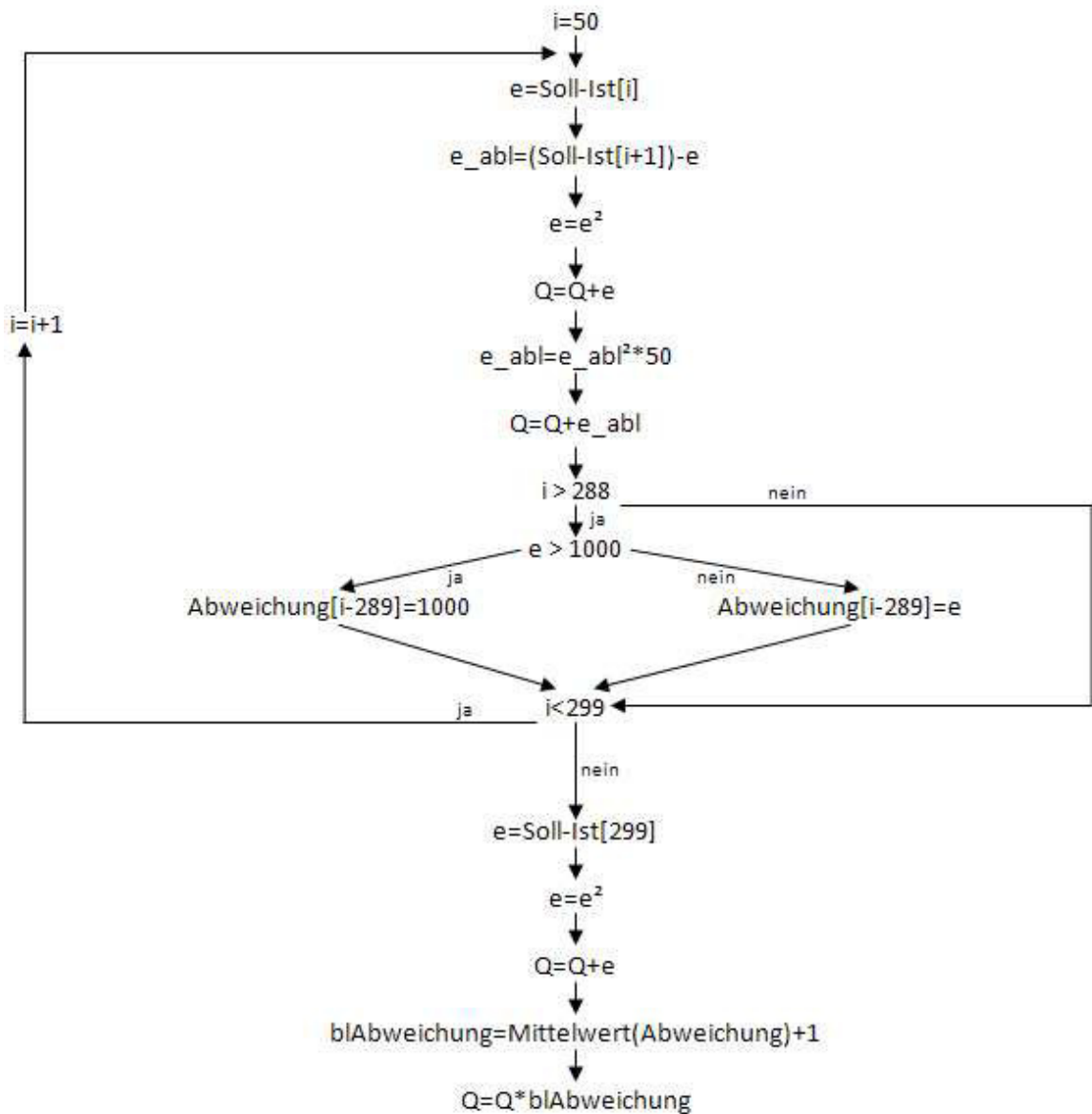
A2: Stateflow Optimierung Opti1 Suchverfahren



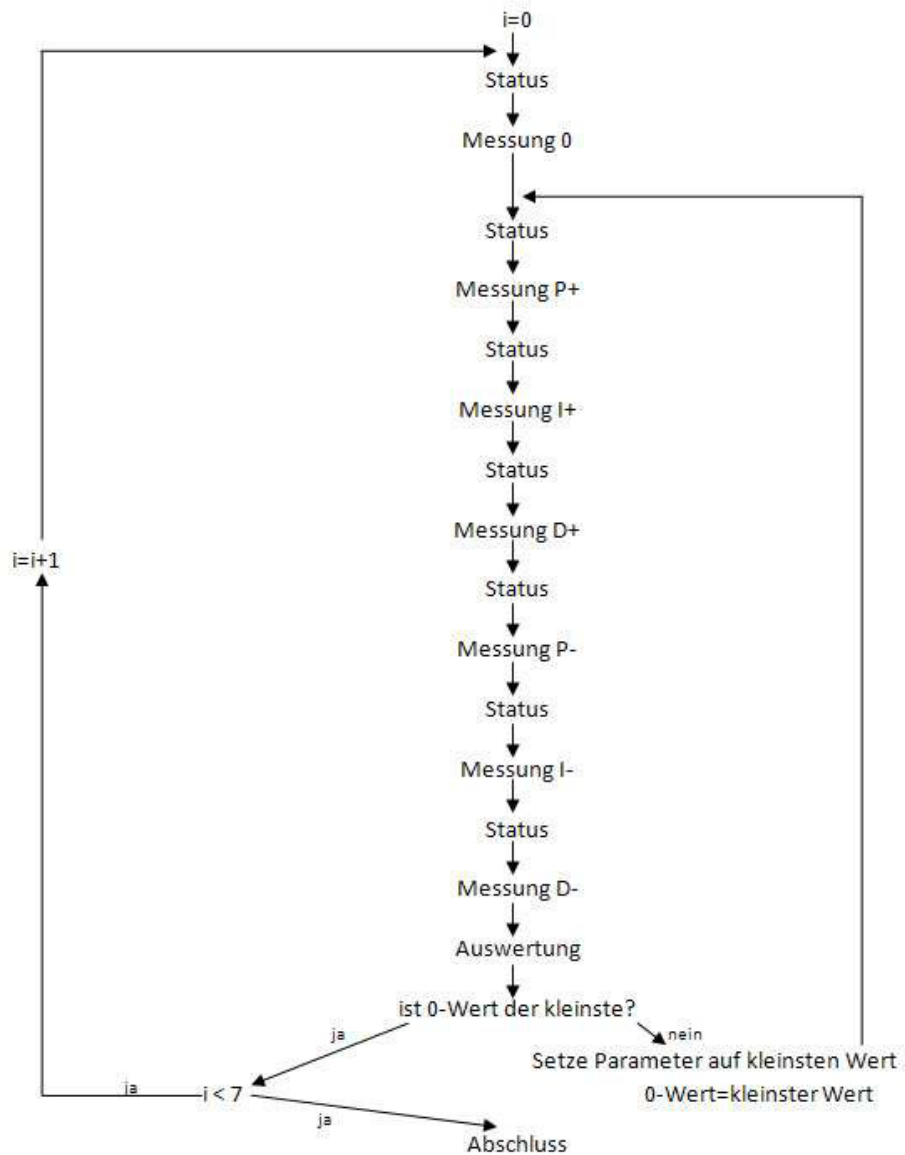
B: Dokumentation zur CAS-Firmware-Erweiterung 1.4

Die Gesamte Dokumentation zur Firmwareerweiterung 1.4 [7] befindet sich auf der beigelegten CD.

B1: Ablaufplan Qualitätserrechnung



B2: Ablaufplan der Optimierung



B3: Quelltextauszug

Firmwareversion 1.4: Zeilen 1601 bis 1651 aus CAS_ChristianThormann.c

```

//Iplus-Messung*****
if(c!=5){
    cli();
    KP[(Pfeife-1)]=P0;
    KI[(Pfeife-1)]=I0+diff[(int)i];
    KD[(Pfeife-1)]=D0;                                //setze PID-Prameter auf Messwert
    Iplus=0;
    sei();
    delay_ms(1);
    ReStEnable=0x01;
    for (j=0; j<sprungzahl; j++){
        t=0;
        t2=0;                                         //Zeitmessvariablen nullen
        cli();
        AutoPara=0x01;                               //Messung initialisieren
        U_soll[(Pfeife-1)]=spruenge[j];              //Sollwert setzen
        sei();
        while(t2==0);                                //Messen (ISR)
        AutoPara=0x00;
        ReStEnable=0x00;                              //Messung beendet
        cli();
        Fhelp=0;
        for(l=50; l<299; l++){                        //50..300ms
            qe=spruenge[j]-sensor[l];                 //e=soll-ist
            qa=(spruenge[j]-sensor[(l+1))]-qe;         //e'=e(t+1)-e(t)
            qe*=qe;                                    //e²
            Fhelp+=qe;                                 //Q=Q+e²
            qa=qa*qa*50;                               //e'=e'*e'*50
            Fhelp+=qa;                                 //Q=Q+e'
            if(l>288){                                //letzten 10 Werte=289..299
                if(qe>1000){                           //wenn abweichung über 31mV (31²=1000)
                    abw[(l-289)]=1000;                 //    abweichung=1000
                }else{
                    abw[(l-289)]=qe;                   //    sonst
                                                        //    abweichung=e²
                }
            }
        }
        qe=spruenge[j]-sensor[299];                  //letzten wert berechnen
        qe*=qe;
        Fhelp+=qe;
        babw=Mittelwert(abw, 10);
        //babw=bleibende Abweichung=Mittelwert der letzten 10 Abweichungen
        babw+=1;
        //+1, babw könnte null sein (aber nicht negativ, da e² genutzt wird)
        Fhelp*=babw;                                  //Q=Q*bleibendeAbweichung
        Iplus+=Fhelp;                                  //Q(gesamtmessung)=Summe der EinzelQ
        sei();
        delay_ms(1);
        ReStEnable=0x01;
    }
}else{
    Iplus=0xFFFFFFFFFFFFFFFF;
}

```


Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 31.01.2013

Christian Thormann

Das Projekt „Pfeifen-Orgel mit dynamischer Stimmung; Ansteuerung der Antriebselemente zur dynamischen Stimmungskorrektur und automatische Selbstkalibration“ wurde durch das BMWi gefördert.

Gefördert durch:



Bundesministerium
für Wirtschaft
und Technologie



aufgrund eines Beschlusses
des Deutschen Bundestages